Scientific Annals of Computer Science vol. 35 (2), 2025, pp. 161–196
doi: 10.47743/SACS.2025.2.161

Conditional Logic as a Short-Circuit Logic

Jan A. Bergstra[1] and Alban Ponse[1]

Abstract

Three-valued conditional logic (CL), defined by Guzmán and Squier (1990) and based on McCarthy's noncommutative connectives, axiomatises a short-circuit logic (SCL), that is, a logic that prescribes short-circuit evaluation of conjunction and disjunction. CL defines more identities than three-valued MSCL (Memorising SCL, which also has a two-valued variant). This follows from the fact that the definable connective that prescribes full left-sequential conjunction is commutative in CL.

We observe that CL also has a two-valued variant of which the full left-sequential connectives and negation define a commutative logic that is weaker than propositional logic because the absorption laws do not hold.

Next, we show that the original, equational axiomatisation of CL is not independent and give several alternative, independent axiomatisations.

Finally, we show that in CL, the full left-sequential connectives and negation define Bochvar's three-valued logic. The paper ends with an appendix on the use of Prover9 and Mace4.

Keywords: Conditional logic, short-circuit evaluation, short-circuit logic, left-sequential connectives, Bochvar's logic.

1 Introduction

Conditional logic (CL) is defined by Guzmán and Squier in the paper *The algebra of conditional logic* [13], where the conjunction adapts McCarthy's

This work is licensed under the Creative Commons Attribution Licence (CC BY)

[1]Informatics Institute, University of Amsterdam, Science Park 900, 1098 XH Amsterdam, The Netherlands, Email: j.a.bergstra@uva.nl & a.ponse@uva.nl

Table 1: The set of laws of CL, given in [13]

(1.1a)	$x'' = x$
(1.1b)	$(x \wedge y)' = x' \vee y'$
(1.1c)	$(x \wedge y) \wedge z = x \wedge (y \wedge z)$
(1.1d)	$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$
(1.1e)	$(x \vee y) \wedge z = (x \wedge z) \vee (x' \wedge y \wedge z)$
(1.1f)	$x \vee (x \wedge y) = x$
(1.1g)	$(x \wedge y) \vee (y \wedge x) = (y \wedge x) \vee (x \wedge y)$

conjunction to the domain of the three truth values *true*, *false*, *undefined* with constants T, F, U, respectively. In CL, the 3-element algebra $C = \{\mathsf{T}, \mathsf{F}, \mathsf{U}\}$ is studied with the operations $x \mapsto x' : C \to C$ and $(x, y) \mapsto x \wedge y, x \vee y : C^2 \to C$ given by the tables

	$'$
T	F
F	T
U	U

$\wedge$	T	F	U
T	T	F	U
F	F	F	F
U	U	U	U

$\vee$	T	F	U
T	T	T	T
F	T	F	U
U	U	U	U

In [13], a *C-algebra* is defined as a certain algebra with the operations $'$, $\wedge$ and $\vee$, and it is shown that the class of C-algebras is a variety. This set-up does not require that the truth constants are defined in a C-algebra, and thus includes four basic cases:

(i) no distinguished elements (truth constants),

(ii) U distinguished,

(iii) T and F distinguished, and

(iv) all elements of C distinguished.

In [13] it is shown that the laws in Table 1 are complete for the variety of C-algebras. Laws (1.1a) and (1.1b) imply that this variety satisfies a duality principle identical to the one in ordinary Boolean algebra. For the case that the constants for these three truth values are distinguished, the following three axioms are added:

$$\mathsf{U}' = \mathsf{U}, \quad \mathsf{T} \wedge x = x, \quad \mathsf{T}' = \mathsf{F}.$$

We end this brief introduction to CL with a quote from [13] about "conditional logic" and short-circuit evaluation:

> "For example, let $B = \{T, F\}$ with ordinary negation ($'$), conjunction ($\wedge$) and disjunction ($\vee$). [...]
> Up to anti-isomorphism, there is a unique regular extension of B to a 3-valued logic with non-commutative $\wedge$ and $\vee$ which satisfies deMorgan's laws $x'' = x$ and $(x \wedge y)' = x' \vee y'$. Following [11], we call this algebra "conditional logic." Conditional logic was first studied by McCarthy [16] and [15]. It is the logic of choice in several programming languages (C, Prolog, Lisp, ...) in which the idea of "short circuit evaluation" is implemented. It leads to faster evaluation of the logical expression, since evaluation stops as soon as an answer can be obtained."
> [Reference numbering has been adjusted.]

We first explain why we view short-circuit evaluation (SCE) as the basic evaluation strategy in any setting that *prescribes* sequential evaluation of $\wedge$ and $\vee$. Following [1], we write

$$x \mathbin{\circ\!\!\wedge} y$$

for the conjunction of x and y that prescribes SCE, where the small circle indicates that the left argument must be evaluated first, and use the name *short-circuit conjunction*. We write $\mathbin{\circ\!\!\vee}$ for *short-circuit disjunction*, which as above is defined dually:

$$x \mathbin{\circ\!\!\vee} y = \neg(\neg x \mathbin{\circ\!\!\wedge} \neg y).$$

From now on, we will mostly use $\mathbin{\circ\!\!\wedge}$ and $\mathbin{\circ\!\!\vee}$ and $\neg$ when writing about CL. We write

$$\mathrm{CL}_2 \quad \text{and} \quad \mathrm{CL}_3$$

respectively for the case where constants T and F are distinguished and the two defining axioms $\neg\mathsf{T} = \mathsf{F}$ and $\mathsf{T} \mathbin{\circ\!\!\wedge} x = x$ are added to those of CL, and the case where U is also distinguished and the axiom $\neg\mathsf{U} = \mathsf{U}$ is added. If no constants are distinguished, we simple write CL.

Another well-known sequential evaluation strategy is *full left-sequential evaluation*, where both arguments of a conjunction and a disjunction are always evaluated. The connective *full left-sequential conjunction*, denoted by $\mathbin{\bullet\!\!\wedge}$, prescribes full left-sequential evaluation and can be defined in terms of SCE:

$$x \mathbin{\bullet\!\!\wedge} y = (x \mathbin{\circ\!\!\vee} (y \mathbin{\circ\!\!\wedge} \mathsf{F})) \mathbin{\circ\!\!\wedge} y.$$

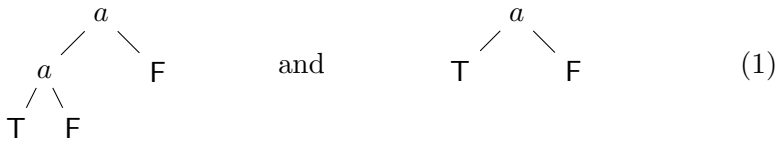
Thus, if x evaluates to *true*, y is evaluated and determines the evaluation value, and if x evaluates to *false*, $y \mathbin{\circ\!\!\wedge} \mathsf{F}$ is evaluated, which determines that $x \mathbin{\bullet\!\!\wedge} y$ has the value *false*. The connective *full left-sequential disjunction*, denoted by $\mathbin{\bullet\!\!\vee}$, is defined dually: $x \mathbin{\bullet\!\!\vee} y = \neg(\neg x \mathbin{\bullet\!\!\wedge} \neg y)$.

Short-circuit evaluation admits several semantics (valuation congruences), which prompted the definition of several equational logics that we introduced in [6] as two-valued *short-circuit logics* (SCLs). For example, $x \mathbin{\owedge} x = x$ is refuted in Free SCL (FSCL), while it holds in Memorising SCL (MSCL). Next, Static SCL (SSCL) is the short-circuit logic that represents propositional logic with connectives $\owedge$ and $\overset{\circ}{\vee}$ instead of $\wedge$ and $\vee$, and thus with short-circuit evaluation as the prescribed evaluation strategy. For example, $x \mathbin{\owedge} y = y \mathbin{\owedge} x$ is refuted in MSCL, while it holds in SSCL.[2]

These SCLs can be ordered by a chain FSCL $\prec$ MSCL $\prec$ SSCL:

$$\begin{aligned} \mathrm{FSCL} \vdash s = t &\implies \mathcal{L} \vdash s = t \quad \text{with } \mathcal{L} \text{ any SCL,} \\ \mathrm{MSCL} \vdash s = t &\implies \mathrm{SSCL} \vdash s = t. \end{aligned}$$

The logic FSCL is immune to side effects: in FSCL, $x \mathbin{\owedge} x = x$ does not hold because a side effect occurring at the first evaluation of an atom (propositional variable), say a, may alter its second evaluation result. We illustrate graphically that $a \mathbin{\owedge} a$ and a are differently evaluated. Consider



(1)

If an occurrence of atom a is evaluated to *true*, the evaluation path descends along the left branch and otherwise along the right branch, and the leaves T and F represent the final evaluation result (either *true* or *false*). Because both evaluation trees are different, $a \mathbin{\owedge} a$ and a are not identified.[3] Evaluation trees and a completeness result for FSCL are discussed in [18].

In the logic MSCL, the identity $x \mathbin{\owedge} x = x$ is true and this represents a setting in which the evaluation of each atom in an expression is memorised and (atomic) side effects cannot occur. Also in MSCL, the entire process of evaluation is distinctive and therefore $\owedge$ is not commutative, e.g., $a \mathbin{\owedge} \mathsf{F}$ requires evaluation of a, while $\mathsf{F} \mathbin{\owedge} a$ does not. Examples of laws of FSCL (and thus of MSCL) are

$$\neg\neg x = x, \quad \mathsf{T} \mathbin{\owedge} x = x, \quad \mathsf{F} \mathbin{\owedge} x = \mathsf{F} \quad \text{and} \quad (x \mathbin{\owedge} y) \mathbin{\owedge} z = x \mathbin{\owedge} (y \mathbin{\owedge} z).$$

[2] In [6], five different, two-valued SCLs are defined, of which FSCL is the weakest and SSCL, a sequential version of propositional logic, is the strongest.

[3] A perhaps more convincing example is obtained if one takes the programming fragment "`(v := v+1) && (v == 1)`" for a. When programming variable `v` has the initial value `0`, $a \mathbin{\owedge} a$ is evaluated to *false*, while a is evaluated to *true*, assuming that `(v := v+1)` always evaluates to *true* and `(v == 1)` evaluates to the test result of `v`'s value and 1.

In MSCL, the connective $\mathbin{\overset{\bullet}{\wedge}}$ does not depend on the presence of the constant F because

$$\mathrm{MSCL} \vdash x \mathbin{\overset{\bullet}{\wedge}} y = (x \mathbin{\overset{\circ}{\wedge}} y) \mathbin{\overset{\circ}{\vee}} (y \mathbin{\overset{\circ}{\wedge}} x).$$

We observe that CL_2 can be seen as a two-valued variant of CL that axiomatises a short-circuit logic that resides in between MSCL and SSCL, which we will define as $\mathrm{C}\ell\mathrm{SCL}_2$ (two-valued Conditional SCL). A simple and interesting example that distinguishes MSCL and $\mathrm{C}\ell\mathrm{SCL}_2$ is the equation

$$x \mathbin{\overset{\bullet}{\wedge}} y = y \mathbin{\overset{\bullet}{\wedge}} x,$$

which holds in the latter and is refuted in MSCL. Because $\mathrm{C}\ell\mathrm{SCL}_2$ refutes $x \mathbin{\overset{\circ}{\wedge}} y = y \mathbin{\overset{\circ}{\wedge}} x$, it follows that $\mathrm{MSCL} \prec \mathrm{C}\ell\mathrm{SCL}_2 \prec \mathrm{SSCL}$.

In [7] we defined three-valued versions of FSCL and MSCL, denoted $\mathrm{FSCL}^{\mathsf{U}}$ and $\mathrm{MSCL}^{\mathsf{U}}$, respectively. CL_3 axiomatises $\mathrm{C}\ell\mathrm{SCL}$, the three-valued version of $\mathrm{C}\ell\mathrm{SCL}_2$, and $\mathrm{MSCL}^{\mathsf{U}} \prec \mathrm{C}\ell\mathrm{SCL}$ follows from the same example. (Of course, SSCL has no extension with U because commutativity of $\mathbin{\overset{\circ}{\wedge}}$ implies $\mathsf{U} = \mathsf{U} \mathbin{\overset{\circ}{\wedge}} \mathsf{F} = \mathsf{F} \mathbin{\overset{\circ}{\wedge}} \mathsf{U} = \mathsf{F}$.)

We continue with a brief introduction to the building blocks for our generic definition of short-circuit logics. In 1985, Hoare introduced in [14] the *conditional*, a ternary connective with notation $x \lhd y \rhd z$ that models **if** y **then** x **else** z, and provided eleven equational axioms to show that the conditional and two constants for truth and falsehood characterise the propositional calculus. With the conditional, algebraic properties can be elegantly expressed, e.g.,

$$\mathsf{T} \lhd x \rhd \mathsf{F} = x \quad \text{and} \quad x \lhd (y \lhd z \rhd u) \rhd v = (x \lhd y \rhd v) \lhd z \rhd (x \lhd u \rhd v).$$

However, the same result was proved in 1948 by Church in [9], where he introduced *conditioned disjunction*,[4] notation $[x, y, z]$, that also models **if** y **then** x **else** z. Moreover, in [9], the duality property of $[x, y, z]$ is highlighted and even older related work is mentioned, a quote:

> "*Conditioned disjunction, t, and f, are a complete set of independent primitive connectives for the propositional calculus.* This has been proved by E. L. Post as a part of his comprehensive study of primitive connectives for the propositional calculus. [footnote: E. L. Post. *The Two-Valued Iterative Systems of Mathematical Logic* (Annals of Mathematics Studies, no. 5), Princeton, N. J., 1941.] But it seems worth

[4] For the conditioned disjunction, reference [10] (1956) is often used, and also the name *conditional disjunction*.

> while to give a separate proof here. [footnote: The method of the proof of completeness is that used by E. L. Post in 1921 in connection with a different set of primitive connectives. See the *American Journal of Mathematics*, vol. 43, pp. 167-168.]"
> [Our comment: the mentioned "method of the proof" is a simple induction with respect to the number of variables of a possible connective.]

Both [14] and [9] do not address the fact that the ternary connective defined therein in fact prescribes short-circuit evaluation.

In [3], we introduced *proposition algebra*, motivated by the observation that Hoare's conditional both naturally prescribes short-circuit evaluation and admits several semantics (valuation congruences), the least distinguishing of which is that of sequential propositional logic. Furthermore,

$$x \mathbin{\diagup\!\!\wedge} y = y \lhd x \rhd \mathsf{F}$$

can be seen as a formal definition of what is usually described as the short-circuit evaluation of conjunction. In [6], we used proposition algebra as an algebraic framework to define several short-circuit logics, including FSCL, MSCL and SSCL, and in [7] to define $\mathrm{FSCL}^{\mathsf{U}}$ and $\mathrm{MSCL}^{\mathsf{U}}$, i.e. the three-valued variants of FSCL and MSCL.

Novel aspects of the paper. We offer the perspective of considering CL as a 'short-circuit logic' (CℓSCL) and place it in a hierarchy of SCLs, each defined using the conditional connective and explicit notations for the sequential, short-circuit connectives. We observe (also) that a two-valued variant exists (C$\ell\mathrm{SCL}_2$) and establish that this is the strongest SCL in the hierarchy of two-valued SCLs below SSCL (the short-circuit version of propositional logic), for example $a \mathbin{\diagup\!\!\wedge} \mathsf{F}$ and F are different for every atom a.

Following [13], CℓSCL is axiomatised by the axioms of $\mathrm{MSCL}^{\mathsf{U}}$ and the axiom $(x \mathbin{\diagup\!\!\wedge} y) \mathbin{\diagdown\!\!\vee} (y \mathbin{\diagup\!\!\wedge} x) = (y \mathbin{\diagup\!\!\wedge} x) \mathbin{\diagdown\!\!\vee} (x \mathbin{\diagup\!\!\wedge} y)$ (cf. (1.1g) in [13]). Inspired by the "reduced trees" in [13] and based on the normalisation functions for MSCL and $\mathrm{MSCL}^{\mathsf{U}}$, we provide normalisation functions for $\mathrm{C}\ell\mathrm{SCL}_2$ and CℓSCL that are both defined with respect to a total order of the set of atoms.

Next, we show that the laws (1.1a)–(1.1g) of [13] are not independent and present independence results (also for the variants with truth constants).

Two more points:

1) We show that in CℓSCL, the definable full left-sequential connectives $\mathbin{\diagup\!\!\bullet\!\!\wedge}$ and $\mathbin{\diagdown\!\!\bullet\!\!\vee}$ together with negation define Bochvar's logic.
2) We briefly discuss an application perspective for CℓSCL.

Contents of the paper.
– In Section 2, we recall the definitions of the proposition algebras CP and CP_{mem} that underlie the definitions of FSCL and MSCL, respectively. We also recall the three-valued variants.
– In Section 3, we define the two-valued proposition algebra CP_{cl} which is related to conditional logic. We define and explain in detail so-called 'CL-basic forms', based on the mem-basic forms for CP_{mem}, which resemble the "reduced trees" defined in [13] that provide a "standard model" for free C-algebras. We prove that CL_2 completely axiomatises $\mathrm{C}\ell\mathrm{SCL}_2$, the SCL defined by CP_{cl} with negation and the short-circuit connectives.
– In Section 4, we define the three-valued proposition algebra $\mathrm{CP}_{cl}^{\mathsf{U}}$ and do the same as in Section 3 for the three-valued variants. It follows that CL_3 is a complete axiomatisation of CℓSCL.
– In Section 5, we show that the laws of CL in Table 1 are not independent and provide several alternatives fot CL, CL_2, CL^{U} (i.e. CL with only U), and CL_3 that are independent.
– In Section 6, we consider the distinguishing example $x \mathbin{\underset{\bullet}{\wedge}} y = y \mathbin{\underset{\bullet}{\wedge}} x$ that proves $\mathrm{MSCL}^{\mathsf{U}} \prec \mathrm{C}\ell\mathrm{SCL}$ and show that in CℓSCL, and thus in CL_3, the (definable) full left-sequential connectives and negation define Bochvar's three-valued logic. We conclude with a comment on the role of the constants T and F in SCLs and some remarks on related work. Finally, we outline an application perspective for CℓSCL.
– The paper ends with two appendices, Appendix A on a normalisation function for mem-basic forms, and Appendix B on the use of the tools *Prover9* and *Mace4* [17].

2 Proposition Algebra and Short-Circuit Logics

In this section we recall the definitions of two-valued and three-valued proposition algebra, and of the short-circuit logics mentioned in the previous section.

Definition 2.1 *For A a set of atoms (propositional variables), define the signatures*

$$\Sigma_{\mathrm{CP}}(A) = \{ _ \lhd _ \rhd _, \mathsf{T}, \mathsf{F}, a \mid a \in A \} \quad \textit{and} \quad \Sigma_{\mathrm{CP}}^{\mathsf{U}}(A) = \Sigma_{\mathrm{CP}}(A) \cup \{\mathsf{U}\}$$

*and the sets $\mathcal{C}_A$ and $\mathcal{C}_A^{\mathsf{U}}$ of **conditional expressions** as the set of closed terms over $\Sigma_{\mathrm{CP}}(A)$ and $\Sigma_{\mathrm{CP}}^{\mathsf{U}}(A)$, respectively.*

Table 2: CP, a set of basic axioms of proposition algebra, and $\mathrm{CP}^{\mathsf{U}} = \mathrm{CP} \cup (\text{CP-U})$

CP :	$x \lhd \mathsf{T} \rhd y = x$	(CP1)
	$x \lhd \mathsf{F} \rhd y = y$	(CP2)
	$\mathsf{T} \lhd x \rhd \mathsf{F} = x$	(CP3)
	$x \lhd (y \lhd z \rhd u) \rhd v = (x \lhd y \rhd v) \lhd z \rhd (x \lhd u \rhd v)$	(CP4)
CP^{U} :	$x \lhd \mathsf{U} \rhd y = \mathsf{U}$	(CP-U)

In Table 2 we display CP, a set of basic axioms for the conditional defined in [3], and CP^{U}, which was defined in [7]. The dual of a conditional expression P, notation P^d, is defined by $\mathsf{T}^d = \mathsf{F}$, $\mathsf{F}^d = \mathsf{T}$, $\mathsf{U}^d = \mathsf{U}$, $a^d = a$ for $a \in A$, and $(P \lhd Q \rhd R)^d = R^d \lhd Q^d \rhd P^d$. Taking the dual of a variable x as itself, it easily follows that CP and CP^{U} are self-dual axiomatisations.

The axioms of CP^{U} define a congruence over $\mathcal{C}_A^{\mathsf{U}}$: *free valuation congruence*, which can be characterised with help of basic forms.[5]

Definition 2.2 ***Basic forms*** *are defined by the following grammar*

$$t ::= \mathsf{T} \mid \mathsf{F} \mid \mathsf{U} \mid t \lhd a \rhd t \quad \textit{for } a \in A.$$

We write BF_A^{U} for the set of basic forms over A, and BF_A for the subset of BF_A^{U} in which U does not occur. The constants in a basic form $P \in BF_A^{\mathsf{U}}$ are called ***leaves****. The* ***depth*** *$d(P)$ of $P \in BF_A^{\mathsf{U}}$ is defined by $d(\mathsf{T}) = d(\mathsf{F}) = d(\mathsf{U}) = 0$ and $d(Q \lhd a \rhd R) = 1 + \max\{d(Q), d(R)\}$.*

In [7, Appendix A8] we prove that for each $P \in \mathcal{C}_A^{\mathsf{U}}$ there is a unique $Q \in BF_A^{\mathsf{U}}$ such that $\mathrm{CP}^{\mathsf{U}} \vdash P = Q$. This implies that for all $P, Q \in \mathcal{C}_A^{\mathsf{U}}$, $\mathrm{CP}^{\mathsf{U}} \vdash P = Q$ if, and only if, P and Q have the same (unique) basic form, that is, if P and Q are free valuation congruent.

[5] About terminology: 1) Two-valued free valuation congruence (FVC) was defined in [3] in terms of "valuation algebras", hence the name. 2) We use "basic form" rather than "normal form" because the "natural" normal form of $\mathsf{T} \lhd a \rhd \mathsf{F}$ would be a, while the basic form of atom a is $\mathsf{T} \lhd a \rhd \mathsf{F}$.

Memorising valuation congruence is defined by $\mathrm{CP}^{\mathsf{U}}_{mem}$, i.e., CP^{U} extended with the axiom

$$x \lhd y \rhd (z \lhd u \rhd (v \lhd y \rhd w)) = x \lhd y \rhd (z \lhd u \rhd w), \qquad \text{(CPmem)}$$

which expresses that the evaluation of y is memorised (see Example 2.6 below). Furthermore, in [6], CP_{mem} is defined as the extension of CP with (CPmem).

By replacing y by $\mathsf{F} \lhd y \rhd \mathsf{T}$ and u by $\mathsf{F} \lhd u \rhd \mathsf{T}$, it follows that

$$\mathrm{CP}^{\mathsf{U}}_{mem} \vdash ((w \lhd y \rhd v) \lhd u \rhd z) \lhd y \rhd z = (w \lhd u \rhd z) \lhd y \rhd x, \qquad \text{(CPmem2)}$$

which is the dual of the above axiom, so the duality principle holds in $\mathrm{CP}^{\mathsf{U}}_{mem}$.

Definition 2.3 *The constants* T, F *and* U *are* ***mem-basic forms****, and a basic form* $P \lhd a \rhd Q$ *is a* ***mem-basic form*** *if* a *does not occur in either* P *or* Q*.*

We write MBF^{U}_A *for the set of mem-basic forms over* A*, and* MBF_A *for the subset of* MBF^{U}_A *in which* U *does not occur.*

Definition 2.4 *For* $a \in A$*, the auxiliary functions* $\ell_a, r_a : BF^{\mathsf{U}}_A \to BF^{\mathsf{U}}_A$*, "left* a*-reduction" and "right* a*-reduction", are defined by*

$$\ell_a(B) = r_a(B) = B \quad \text{for } B \in \{\mathsf{T}, \mathsf{F}, \mathsf{U}\},$$

$$\ell_a(P \lhd b \rhd Q) = \begin{cases} P & \text{if } b = a, \\ \ell_a(P) \lhd b \rhd \ell_a(Q) & \text{if } b \neq a, \end{cases}$$

$$r_a(P \lhd b \rhd Q) = \begin{cases} Q & \text{if } b = a, \\ r_a(P) \lhd b \rhd r_a(Q) & \text{if } b \neq a. \end{cases}$$

So, for each $P \in MBF^{\mathsf{U}}_A$, both $\ell_a(P)$ and $r_a(P)$ are mem-basic forms that do not contain a. Mem-basic forms and the functions $\ell_a()$ and $r_a()$ are a point of departure in the next section.

In [4, Thm.5.7] we proved a result that implies that for each $P \in \mathcal{C}_A$, there is a unique $Q \in MBF_A$ such that $\mathrm{CP}_{mem} \vdash P = Q$, and for all $P, Q \in \mathcal{C}_A$, $\mathrm{CP}_{mem} \vdash P = Q$ if, and only if, P and Q have the same (unique) mem-basic form, that is, if P and Q are memorising valuation congruent. Given the approach in [7], the same results follow for the three-valued setting.

Definition 2.5 *For* A *a set of atoms (propositional variables), define the signatures*

$$\Sigma_{\mathrm{SCL}}(A) = \{\owedge, \overset{\circ}{\vee}, \neg, \mathsf{T}, \mathsf{F}, a \mid a \in A\} \quad \text{and} \quad \Sigma^{\mathsf{U}}_{\mathrm{SCL}}(A) = \Sigma_{\mathrm{SCL}}(A) \cup \{\mathsf{U}\}$$

and the sets $\mathcal{S}_A$ *and* $\mathcal{S}^{\mathsf{U}}_A$ *of* ***sequential expressions*** *as the set of closed terms over* $\Sigma_{\mathrm{SCL}}(A)$ *and* $\Sigma^{\mathsf{U}}_{\mathrm{SCL}}(A)$*, respectively.*

The short-circuit connectives are defined in CP^{U} (and in CP) by the following axioms:

$$\neg x = \mathsf{F} \lhd x \rhd \mathsf{T}, \tag{defNeg}$$

$$x \mathbin{\owedge} y = y \lhd x \rhd \mathsf{F}, \tag{defAnd}$$

and, by duality, short-circuit disjunction should satisfy in CP^{U} (and in CP)

$$x \mathbin{\overline{\vee}} y = \mathsf{T} \lhd x \rhd y, \tag{defOr}$$

which we show below. In a similar way it can be shown that full left-sequential conjunction, which we already defined by $x \mathbin{\bullet\!\!\wedge} y = (x \mathbin{\overline{\vee}} (y \mathbin{\owedge} \mathsf{F})) \mathbin{\owedge} y$, satisfies

$$x \mathbin{\bullet\!\!\wedge} y = y \lhd x \rhd (\mathsf{F} \lhd y \rhd \mathsf{F}).$$

Example 2.6 *Consider the evaluation trees of $a \mathbin{\owedge} a$ and a in* (1) *(in the Introduction), i.e. those of $(\mathsf{T} \lhd a \rhd \mathsf{F}) \lhd a \rhd \mathsf{F}$ and a, i.e. $\mathsf{T} \lhd a \rhd \mathsf{F}$. According to consequence* (CPmem2) *with $u = \mathsf{T}$, $\mathrm{CP}^{\mathsf{U}}_{mem} \vdash (w \lhd y \rhd v) \lhd y \rhd z = w \lhd y \rhd x$, so $(\mathsf{T} \lhd a \rhd \mathsf{F}) \lhd a \rhd \mathsf{F} = \mathsf{T} \lhd a \rhd \mathsf{F}$. Hence, the evaluation of the first a in $(\mathsf{T} \lhd a \rhd \mathsf{F}) \lhd a \rhd \mathsf{F}$ is "memorised" and the second a needs no evaluation.*

In [6] we defined (two-valued) short-circuit logics in a generic way with help of the conditional by restricting the consequences of a particular CP-axiomatisation extended with equations (defNeg) and (defAnd) to the signature $\Sigma_{\mathrm{SCL}}(A)$, and we repeat here some of these definitions. So, the conditional connective is considered a hidden operator. In these definitions, the export operator $\square$ of Module algebra [2] is used to express hiding in a concise way: in module algebra, $S \square X$ is the operation that exports the signature S from module X while declaring other signature elements hidden.

Definition 2.7 *A **short-circuit logic** is a logic that implies the consequences of the module expression*

$$\mathrm{SCL} = \{\mathsf{T}, \neg, \mathbin{\owedge}\} \square (\mathrm{CP} \cup \{(\mathrm{defNeg}), (\mathrm{defAnd})\}).$$

Definition 2.8 ***Free short-circuit logic*** (**FSCL**) *is the short-circuit logic that implies no other consequences than those of the module expression* SCL.

Memorising short-circuit logic (**MSCL**) *is the short-circuit logic that implies no other consequences than those of the module expression*

$$\{\mathsf{T}, \neg, \mathbin{\owedge}\} \square (\mathrm{CP}_{mem} \cup \{(\mathrm{defNeg}), (\mathrm{defAnd})\}).$$

Static short-circuit logic (**SSCL**) *is the short-circuit logic that implies no other consequences than those of the module expression*

$$\{\mathsf{T}, \neg, \mathbin{\owedge}\} \square (\mathrm{CP}_{mem} \cup \{\mathsf{F} \lhd x \rhd \mathsf{F} = \mathsf{F}, (\mathrm{defNeg}), (\mathrm{defAnd})\}).$$

Table 3: The set EqMSCL of axioms for MSCL

$\mathsf{F} = \neg\mathsf{T}$	(Neg)
$x \mathbin{\overset{\circ}{\vee}} y = \neg(\neg x \mathbin{\overset{\circ}{\wedge}} \neg y)$	(Or)
$\mathsf{T} \mathbin{\overset{\circ}{\wedge}} x = x$	(Tand)
$x \mathbin{\overset{\circ}{\wedge}} (x \mathbin{\overset{\circ}{\vee}} y) = x$	(Abs)
$(x \mathbin{\overset{\circ}{\vee}} y) \mathbin{\overset{\circ}{\wedge}} z = (\neg x \mathbin{\overset{\circ}{\wedge}} (y \mathbin{\overset{\circ}{\wedge}} z)) \mathbin{\overset{\circ}{\vee}} (x \mathbin{\overset{\circ}{\wedge}} z)$	(Mem)

To enhance readability, we extend these short-circuit logics with the constant F and its defining axiom $\mathsf{F} = \neg\mathsf{T}$, which is justified by the SCL-derivation

$$\begin{aligned} \mathsf{F} &= \mathsf{F} \lhd \mathsf{T} \rhd \mathsf{T} && \text{by (CP1)} \\ &= \neg\mathsf{T}, && \text{by (defNeg)} \end{aligned}$$

and with the connective $\mathbin{\overset{\circ}{\vee}}$ and its defining axiom $x \mathbin{\overset{\circ}{\vee}} y = \neg(\neg x \mathbin{\overset{\circ}{\wedge}} \neg y)$, which is justified by

$$\begin{aligned} &\neg(\neg x \mathbin{\overset{\circ}{\wedge}} \neg y) \\ &\quad = \mathsf{F} \lhd ((\mathsf{F} \lhd y \rhd \mathsf{T}) \lhd (\mathsf{F} \lhd x \rhd \mathsf{T}) \rhd \mathsf{F}) \rhd \mathsf{T} && \text{by (defNeg), (defAnd)} \\ &\quad = \mathsf{F} \lhd (\mathsf{F} \lhd x \rhd (\mathsf{F} \lhd y \rhd \mathsf{T})) \rhd \mathsf{T} && \text{by (CP4), (CP2), (CP1)} \\ &\quad = (\mathsf{F} \lhd \mathsf{F} \rhd \mathsf{T}) \lhd x \rhd (\mathsf{F} \lhd (\mathsf{F} \lhd y \rhd \mathsf{T}) \rhd \mathsf{T}) && \text{by (CP4)} \\ &\quad = \mathsf{T} \lhd x \rhd y. && \text{by (CP2), (CP4), (CP1)} \end{aligned}$$

The equational logic EqMSCL is defined by the five axioms in Table 3. In [7] it is proved that EqMSCL axiomatises MSCL. Some consequences of EqMSCL are these (see [7]):

(i) the connective $\mathbin{\overset{\circ}{\wedge}}$ is associative, i.e. $(x \mathbin{\overset{\circ}{\wedge}} y) \mathbin{\overset{\circ}{\wedge}} z = x \mathbin{\overset{\circ}{\wedge}} (y \mathbin{\overset{\circ}{\wedge}} z)$,

(ii) the connective $\mathbin{\overset{\circ}{\wedge}}$ is idempotent, i.e. $x \mathbin{\overset{\circ}{\wedge}} x = x$,

(iii) the connective $\mathbin{\overset{\circ}{\wedge}}$ is left-distributive, i.e. $x \mathbin{\overset{\circ}{\wedge}} (y \mathbin{\overset{\circ}{\vee}} z) = (x \mathbin{\overset{\circ}{\wedge}} y) \mathbin{\overset{\circ}{\vee}} (x \mathbin{\overset{\circ}{\wedge}} z)$, and

(iv) $\neg\neg x = x$, $x \mathbin{\overset{\circ}{\wedge}} \neg x = \neg x \mathbin{\overset{\circ}{\wedge}} x$, and $x \mathbin{\overset{\circ}{\wedge}} (y \mathbin{\overset{\circ}{\wedge}} x) = x \mathbin{\overset{\circ}{\wedge}} y$.

An important property of MSCL is that the conditional can be expressed:

$$
\begin{aligned}
&(x \mathbin{\owedge} y) \mathbin{\overset{\circ}{\vee}} (\neg x \mathbin{\owedge} z)\\
&\quad = \mathsf{T} \lhd (y \lhd x \rhd \mathsf{F}) \rhd (\mathsf{F} \lhd x \rhd z) && \text{by (defNeg), (defOr), (CP2),}\\
&\qquad && \text{and (CP4)}\\
&\quad = (\mathsf{T} \lhd y \rhd (\mathsf{F} \lhd x \rhd z)) \lhd x \rhd (\mathsf{F} \lhd x \rhd z) && \text{by (CP4), (CP2)}\\
&\quad = (\mathsf{T} \lhd y \rhd \mathsf{F}) \lhd x \rhd (u \lhd \mathsf{F} \rhd (\mathsf{F} \lhd x \rhd z)) && \text{by (CPmem), (CP2)}\\
&\quad = y \lhd x \rhd z. && \text{by (CP3), (CPmem), (CP2)}
\end{aligned}
$$

In [7], we also proved

$$
\begin{aligned}
\mathrm{EqMSCL} \vdash (x \mathbin{\owedge} y) \mathbin{\overset{\circ}{\vee}} (\neg x \mathbin{\owedge} z) &= (\neg x \mathbin{\owedge} z) \mathbin{\overset{\circ}{\vee}} (x \mathbin{\owedge} y)\\
&= (x \mathbin{\overset{\circ}{\vee}} z) \mathbin{\owedge} (\neg x \mathbin{\overset{\circ}{\vee}} y).
\end{aligned}
$$

Finally, we defined in [7] the short-circuit logics $\mathrm{FSCL}^{\mathsf{U}}$ and $\mathrm{MSCL}^{\mathsf{U}}$. We proved that MSCL and $\mathrm{MSCL}^{\mathsf{U}}$ are axiomatised by the equational logics EqMSCL and $\mathrm{EqMSCL}^{\mathsf{U}} = \mathrm{EqMSCL} \cup \{\neg \mathsf{U} = \mathsf{U}\}$, respectively.

3 Conditional Short-Circuit Logic, the Two-Valued Case

In this section we restrict to the two-valued setting. In particular, where we write about mem-basic forms, we refer to those in MBF_A (see Definition 2.3). We define 'CL-basic forms' based on the mem-basic forms for CP_{mem} and the short-circuit logic $\mathrm{C}\ell\mathrm{SCL}_2$ and prove that CL_2 completely axiomatises $\mathrm{C}\ell\mathrm{SCL}_2$, the two-valued SCL defined by CP_{cl} and the short-circuit connectives.

Consider the axiom

$$\mathsf{T} \lhd (y \lhd x \rhd \mathsf{F}) \rhd (x \lhd y \rhd \mathsf{F}) = \mathsf{T} \lhd (x \lhd y \rhd \mathsf{F}) \rhd (y \lhd x \rhd \mathsf{F}), \tag{CL1}$$

i.e. the translation of the CL-axiom $(x \mathbin{\owedge} y) \mathbin{\overset{\circ}{\vee}} (y \mathbin{\owedge} x) = (y \mathbin{\owedge} x) \mathbin{\overset{\circ}{\vee}} (x \mathbin{\owedge} y)$ to CP, and the following axiom that is more convenient for a semantics based on mem-basic forms:

$$(x \lhd y \rhd z) \lhd u \rhd (v \lhd y \rhd w) = (x \lhd u \rhd v) \lhd y \rhd (z \lhd u \rhd w). \tag{CL2}$$

Lemma 3.1 (i) $\mathrm{CP}_{mem} + (\mathrm{CL1}) \vdash (\mathrm{CL2})$, *and* (ii) $\mathrm{CP}_{mem} + (\mathrm{CL2}) \vdash (\mathrm{CL1})$.

Proof: (i) Derive

$$\begin{aligned} x \lhd (\mathsf{T} \lhd (y \lhd u \rhd \mathsf{F}) \rhd (u \lhd y \rhd \mathsf{F})) \rhd v \qquad (2)\\ = x \lhd (y \lhd u \rhd \mathsf{F}) \rhd (x \lhd (u \lhd y \rhd \mathsf{F}) \rhd v)\\ = (x \lhd y \rhd (x \lhd (u \lhd y \rhd \mathsf{F}) \rhd v)) \lhd u \rhd (x \lhd (u \lhd y \rhd \mathsf{F}) \rhd v)\\ = (x \lhd y \rhd ((x \lhd u \rhd v) \lhd y \rhd v)) \lhd u \rhd ((x \lhd u \rhd v) \lhd y \rhd v)\\ = (x \lhd y \rhd v) \lhd u \rhd (v \lhd y \rhd v), \qquad (3) \end{aligned}$$

and apply axiom (CL1) in (2), so exchange u and y. By (3),

$$(x \lhd y \rhd v) \lhd u \rhd (v \lhd y \rhd v) = (x \lhd u \rhd v) \lhd y \rhd (v \lhd u \rhd v).$$

Now replace v by $v \lhd y \rhd (z \lhd u \rhd w)$ and apply memorisation:

$$(x \lhd y \rhd z) \lhd u \rhd (v \lhd y \rhd w) = (x \lhd u \rhd v) \lhd y \rhd (z \lhd u \rhd w).$$

(ii) Derive $\mathsf{T} \lhd (y \lhd x \rhd \mathsf{F}) \rhd (x \lhd y \rhd \mathsf{F}) = (\mathsf{T} \lhd y \rhd \mathsf{F}) \lhd x \rhd (\mathsf{F} \lhd y \rhd \mathsf{F})$ and apply (CL2). □

Definition 3.2 *Let* CP_{cl} *be the extension of* CP_{mem} *with axiom* (CL2). ***Conditional short-circuit logic*** ($\mathbf{C}\ell\mathbf{SCL}_2$) *is the short-circuit logic that implies no other consequences than those of the module expression*

$$\{\mathsf{T}, \neg, \mathbin{\diamond\!\!\wedge}\} \,\square\, (\mathrm{CP}_{cl} \cup \{(\mathrm{defNeg}), (\mathrm{defAnd})\}).$$

Observe that the axiom (CL2) is self-dual, so the duality principle holds in CP_{cl}. In the setting of CL, we shall adopt a strict total ordering $<$ of A, in order to deal with the axiom (CL2). Whenever we want to be explicit about this ordering, we write

$$(A, <).$$

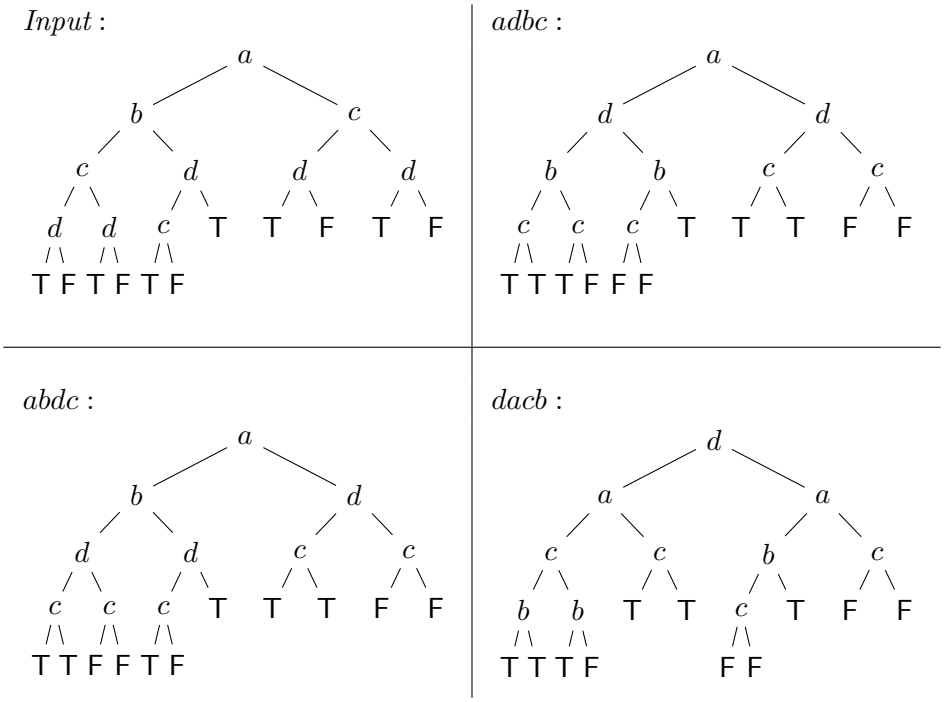
Below we define CL-basic forms and prove that $\mathrm{CP}_{cl} \vdash P = Q$ if, and only if, P and Q are *conditional valuation congruent*, that is, if P and Q have the same unique CL-basic form (relative to a strict total ordering of A).

Example 3.3 *Consider the mem-basic form* $Input = P \lhd a \rhd Q$ *with*

$$\begin{aligned} P &= ((\mathsf{T} \lhd d \rhd \mathsf{F}) \lhd c \rhd (\mathsf{T} \lhd d \rhd \mathsf{F})) \lhd b \rhd ((\mathsf{T} \lhd c \rhd \mathsf{F}) \lhd d \rhd \mathsf{T}),\\ Q &= (\mathsf{T} \lhd d \rhd \mathsf{F}) \lhd c \rhd (\mathsf{T} \lhd d \rhd \mathsf{F}). \end{aligned}$$

Below, Input is shown graphically as the tree at the top left. Furthermore, CP_{cl}*-equivalent representations of Input, with root atoms* a *and* d*, respectively,*

are shown in the right column, and another with root atom a and $d < c$ is shown at the bottom left:



This example is a first step to our definition of so-called CL-basic forms*, a proper subset of the mem-basic forms.*

We discuss the question of how mem-basic forms can be assigned a unique representation modulo conditional valuation congruence, where this representation itself must of course be a mem-basic form. In the following recursive definition, the "shared alphabet" of a mem-basic form is the set of atoms that occur in *each* evaluation path (whereas the "alphabet" $\alpha(P)$ of a mem-basic form P is the set of all its atoms, i.e. those that occur in at least one evaluation path).

Definition 3.4 *For a mem-basic form P, its* ***shared alphabet*** $\widetilde{\alpha}(P)$ *is the set defined by*

$$\widetilde{\alpha}(\mathsf{T}) = \widetilde{\alpha}(\mathsf{F}) = \emptyset, \quad \widetilde{\alpha}(P_1 \lhd a \rhd P_2) = \{a\} \cup (\widetilde{\alpha}(P_1)) \cap \widetilde{\alpha}(P_2)).$$

Hence, $\widetilde{\alpha}(P) = \emptyset$ iff $P \in \{\mathsf{T}, \mathsf{F}\}$, and if $P = P_1 \lhd a \rhd P_2$, then $a \in \widetilde{\alpha}(P)$. In Example 3.3, $\widetilde{\alpha}(\mathit{Input}) = \{a, d\}$, and this is also the shared alphabet of the other mem-basic forms.

Lemma 3.5 *If P is a mem-basic form and $a \in \widetilde{\alpha}(P)$, then* $\mathrm{CP}_{cl} \vdash P = \ell_a(P) \lhd a \rhd r_a(P)$.

Proof: By structural induction on P. If $P = P_1 \lhd a \rhd P_2$, then $\ell_a(P) = P_1$ and $r_a(P) = P_2$, so we are done, and if $P = P_1 \lhd b \rhd P_2$ with $b \neq a$, then $a \in (\widetilde{\alpha}(P_1) \cap \widetilde{\alpha}(P_2))$, hence

$$\begin{aligned}
&\mathrm{CP}_{cl} \vdash P_1 \lhd b \rhd P_2 \\
&\quad = (\ell_a(P_1) \lhd a \rhd r_a(P_1)) \lhd b \rhd (\ell_a(P_2) \lhd a \rhd r_a(P_2)) \quad \text{(by induction)} \\
&\quad = (\ell_a(P_1) \lhd b \rhd \ell_a(P_2)) \lhd a \rhd (r_a(P_1) \lhd b \rhd r_a(P_2)) \quad \text{(by axiom (CL2))} \\
&\quad = \ell_a(P) \lhd a \rhd r_a(P).
\end{aligned}$$

□

Lemma 3.5 implies the following consequence, which shows that $\widetilde{\alpha}()$ is preserved under provable equality:

Lemma 3.6 *If P, Q are mem-basic forms and* $\mathrm{CP}_{cl} \vdash P = Q$, *then* $\widetilde{\alpha}(P) = \widetilde{\alpha}(Q)$.

Proof: If $a \in \widetilde{\alpha}(P)$, then $\mathrm{CP}_{cl} \vdash P = \ell_a(P) \lhd a \rhd r_a(P)$, so $\mathrm{CP}_{cl} \vdash Q = \ell_a(P) \lhd a \rhd r_a(P)$ and thus $a \in \widetilde{\alpha}(Q)$. By symmetry, $\widetilde{\alpha}(P) = \widetilde{\alpha}(Q)$. □

To decide whether different mem-basic forms are provably equal in CP_{cl}, we define so-called CL-basic forms in a recursive way. We first define some technical notions that are used in these basic forms.

Definition 3.7 *For A a set of atoms, A^s is the set of words over A with the property that each $\sigma \in A^s$ contains no multiple occurrences of the same atom. We write ϵ for the empty string, thus $\epsilon \in A^s$.*

For $\sigma \in A^s$ with length $|\sigma| = n$ and $P_0, ..., P_{2^n-1}$ mem-basic forms, define $F_\sigma(P_0, ..., P_{2^n-1})$ by

$$\begin{aligned}
&\sigma = \epsilon : && F_\sigma(P_0) = P_0, \\
&\sigma = a\rho,\ a \in A : && F_\sigma(P_0, ..., P_{2^n-1}) = \\
& && F_\rho(P_0, ..., P_{2^{|\rho|}-1}) \lhd a \rhd F_\rho(P_{2^{|\rho|}}, ..., P_{2^n-1}).
\end{aligned}$$

We say that $\sigma \in A^s$ ***agrees with*** *$(A, <)$ if σ's atoms occur in increasing order.*

For A' a finite set of atoms, we say that the unique string $\rho \in (A')^s$ that contains all atoms of A' in increasing order ***represents*** *$(A', <)$ (thus $|\rho| = |A'|$).*

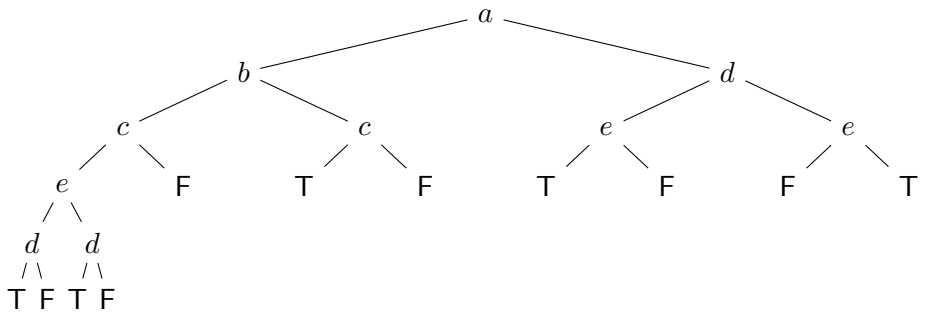
Definition 3.8 *A mem-basic form $P \in MBF_A$ is a **CL-basic form** if for σ that represents $\widetilde{\alpha}(P)$,*

$$P = F_\sigma(P_0, ..., P_{2^{|\sigma|}-1}) \text{ and each } P_i \text{ is a CL-basic form.}$$

We will often just write **Cbf** instead of "CL-basic form". Observe that a mem-basic form P is a Cbf if axiom (CL2) cannot be applied to P (i.e. if $|\widetilde{\alpha}(Q)| \leq 1$ for all subterms Q of P), e.g. as in $(\mathsf{T} \lhd b \rhd \mathsf{F}) \lhd a \rhd (\mathsf{T} \lhd c \rhd \mathsf{F})$.

In Example 3.3, the terms displayed in the right column are Cbfs if *adbc* agrees with $(A, <)$ or if *dacb* does, respectively, and those in the left column are not because they do not satisfy the condition $P = F_{ad}(...)$ or $P = F_{da}(...)$, respectively. We give another example that further illustrates the role of $(A, <)$ and provides more intuition about the existence of unique Cbfs.

Example 3.9 *Consider the following mem-basic form P, which we display graphically:*



Clearly, P is a mem-basic form and $\widetilde{\alpha}(P) = \{a\}$. Observe that some subterms have overlap and can be rearranged according to axiom (CL2)*.*

To obtain a unique CL-basic form, the two outer subterms

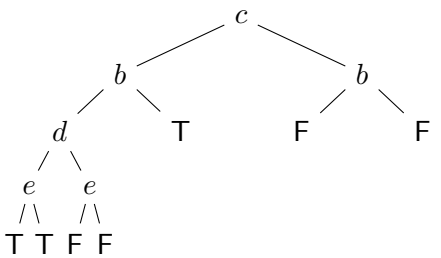
$$(\mathsf{T} \lhd d \rhd \mathsf{F}) \lhd e \rhd (\mathsf{T} \lhd d \rhd \mathsf{F}) \text{ and } (\mathsf{T} \lhd e \rhd \mathsf{F}) \lhd d \rhd (\mathsf{T} \lhd e \rhd \mathsf{F})$$

should be made uniform and it must be decided whether b and the two adjacent c's should be interchanged (with rearranged subterms) or not.

If abcde agrees with $(A, <)$, the left subterm $(\mathsf{T} \lhd d \rhd \mathsf{F}) \lhd e \rhd (\mathsf{T} \lhd d \rhd \mathsf{F})$ should be replaced by

$$(\mathsf{T} \lhd e \rhd \mathsf{T}) \lhd d \rhd (\mathsf{F} \lhd e \rhd \mathsf{F}),$$

and if acbde agrees with $(A, <)$, the left subterm $Q \lhd b \rhd R$ should be replaced by one with root c that we display graphically:



Note that each alternative ordering $(A, <')$ determines how P should be transformed into a provably equal Cbf because it determines whether $b <' c$ or $c <' b$ and $d <' e$ or $e <' d$, and that a is the only possible root atom of such a transformation.

For P a mem-basic form different from T or F, a consequence of axiom (CL2) concerns a provably equal representation ordered by $\widetilde{\alpha}(P)$.

Lemma 3.10 *For each mem-basic form $P \in MBF_A$ and σ that represents $\widetilde{\alpha}(P)$ there exist mem-basic forms $P_0, ..., P_{2^{|\sigma|}-1}$ such that*

$$\mathrm{CP}_{cl} \vdash P = F_\sigma(P_0, ..., P_{2^{|\sigma|}-1}).$$

Proof: By induction on the depth of P. The base case $d(P) = 0$ is trivial. If $d(P) \geq 1$ and $\sigma = a\rho$, then ρ represents the ordering of the shared alphabets of the mem-basic forms $\ell_a(P)$ and $r_a(P)$ that are both of lesser depth, hence

$$\begin{aligned} \mathrm{CP}_{cl} \vdash P &= \ell_a(P) \lhd a \rhd r_a(P) && \text{(by Lemma 3.5)} \\ &= F_\rho(P_0, ..., P_{2^{|\rho|}-1}) \lhd a \rhd F_\rho(P'_0, ..., P'_{2^{|\rho|}-1}) && \text{(IH)} \\ &= F_\sigma(P_0, ..., P_{2^{|\sigma|}-1}) \end{aligned}$$

if $P_{2^{|\rho|}+i} = P'_i$ for $i = 0, ..., 2^{|\rho|} - 1$. □

Theorem 3.11 *For each mem-basic form $P \in MBF_A$ there is a unique CL-basic form Q such that $\mathrm{CP}_{cl} \vdash P = Q$.*

Proof: By induction on the depth of P. If $d(P) = 0$ then $P \in \{\mathsf{T}, \mathsf{F}\}$ and we are done.

If $d(P) > 0$ then $|\widetilde{\alpha}(P)| = n$ for some $n \geq 1$. For σ that represents $\widetilde{\alpha}(P)$ it follows by Lemma 3.10 that $\mathrm{CP}_{cl} \vdash P = F_\sigma(P_0, ..., P_{2^n-1})$. By induction, there are unique Cbfs Q_i such that $\mathrm{CP}_{cl} \vdash P_i = Q_i$. Hence, $Q = F_\sigma(Q_0, ..., Q_{2^n-1})$ is the unique Cbf that satisfies $\mathrm{CP}_{cl} \vdash P = Q$. □

In [4, Thm.5.7] we prove that for each $P \in \mathcal{C}_A$, there is a unique mem-basic form obtained by a normalisation function $mbf()$ such that $\mathrm{CP}_{mem} \vdash P = mbf(P)$. Hence, we find the following result:

Corollary 3.12 *For all $P, Q \in \mathcal{C}_A$, $\mathrm{CP}_{cl} \vdash P = Q$ if, and only if,*

$$[\![P]\!]_{cl} = [\![Q]\!]_{cl}, \textit{ where } [\![P]\!]_{cl} \textit{ is } P\textit{'s unique CL-basic form.}$$

Corollary 3.12 and the above theorem imply that CP_{cl} axiomatises two-valued conditional valuation congruence, that is, the congruence defined by having equal CL-basic forms.

It remains to be shown that $\mathrm{C}\ell\mathrm{SCL}_2$ is completely axiomatised by the equational logic CL_2. Given a signature Σ, we write $\mathbb{T}_{\Sigma,\mathcal{X}}$ for the set of open terms over Σ with variables in $\mathcal{X}$. As in [7], we define the following translation functions:

$f : \mathbb{T}_{\Sigma_{\mathbf{SCL}}(A),\mathcal{X}} \to \mathbb{T}_{\Sigma_{\mathbf{CP}}(A),\mathcal{X}}$ by

$$\begin{aligned} f(B) &= B \ \text{ for } B \in \{\mathsf{T},\mathsf{F}\}, & f(\neg t) &= \mathsf{F} \lhd f(t) \rhd \mathsf{T},\\ f(a) &= a \ \text{ for } a \in A, & f(t_1 \mathbin{\owedge} t_2) &= f(t_2) \lhd f(t_1) \rhd \mathsf{F},\\ f(x) &= x \ \text{ for } x \in \mathcal{X}, & f(t_1 \mathbin{\overline{\vee}} t_2) &= \mathsf{T} \lhd f(t_1) \rhd f(t_2). \end{aligned}$$

$g : \mathbb{T}_{\Sigma_{\mathbf{CP}}(A),\mathcal{X}} \to \mathbb{T}_{\Sigma_{\mathbf{SCL}}(A),\mathcal{X}}$ by

$$g(B) = B \ \text{ for } B \in \{\mathsf{T},\mathsf{F}\},\ g(x) = x \ \text{ for } x \in \mathcal{X},\ g(a) = a \ \text{ for } a \in A,$$
$$g(t_1 \lhd t_2 \rhd t_3) = (g(t_2) \mathbin{\owedge} g(t_1)) \mathbin{\overline{\vee}} (\neg g(t_2) \mathbin{\owedge} g(t_3)).$$

Theorem 3.13 *For all terms s, t over $\Sigma_{\mathrm{SCL}}(A)$, $\mathrm{CL}_2 \vdash s = t \iff \mathrm{C}\ell\mathrm{SCL}_2 \vdash s = t$.*

Proof: In [7, Thm.6.5] it is proved that for all terms s, t over $\Sigma_{\mathrm{SCL}}(A)$,

$$\mathrm{EqMSCL} \vdash s = t \iff \mathrm{MSCL} \vdash s = t$$

and the two crucial intermediate results in this proof, when lifted to the present case, are
1) for all $s, t \in \mathbb{T}_{\Sigma_{\mathrm{CP}}(A),\mathcal{X}}$, $\mathrm{CP}_{cl} \vdash s = t \ \Rightarrow\ \mathrm{CL}_2 \vdash g(s) = g(t)$, and
2) for all $t \in \mathbb{T}_{\Sigma_{\mathrm{SCL}}(A),\mathcal{X}}$, $\mathrm{CL}_2 \vdash g(f(t)) = t$.

For result 1 it must be shown that the axioms of CL_2 are derivable from $\mathrm{C}\ell\mathrm{SCL}_2$. By Theorem 5.2, $\mathrm{CL}_2 \vdash \mathrm{EqMSCL}$, so it remains to be shown that axiom (CL2) is derivable from CL_2, and this follows from Lemma 3.1

because axiom (CL1) is the $g()$-translation of axiom (1.1g). Result 2 holds for EqMSCL, and hence for CL_2. □

Finally, we can use CL-basic forms to define "normal forms" in the equational theory of CL_2: [6] for $P \in \mathcal{S}_A$ we find $f(P) \in \mathcal{C}_A$, and thus, given an ordering $(A, <)$, a unique CL-basic form $mbf(f(P))$ such that $\mathrm{CP}_{cl} \vdash f(P) = mbf(f(P))$ and thus

$$\mathrm{CL}_2 \vdash P = g(f(P)) = g(mbf(f(P))),$$

so $g(mbf(f(P)))$ can be defined as the normal form of P.

4 Conditional Short-Circuit Logic, the Three-Valued Case

In this section we consider the three-valued setting and start from mem-basic forms that may contain U, thus the set MBF_A^{U} of mem-basic forms. We proceed as in Section 3 and it will appear that the CL-basic forms with U are a bit more complex.

Definition 4.1 *Let* $\mathrm{CP}_{cl}^{\mathsf{U}}$ *be the extension of* CP_{cl} *with axiom* (CP-U)*, i.e.* $x \lhd \mathsf{U} \rhd y = \mathsf{U}$.

Conditional short-circuit logic with undefinedness (**CℓSCL**) *is the short-circuit logic that implies no other consequences than those of the module expression*

$$\{\mathsf{T}, \mathsf{U}, \neg, \curlywedge\} \square (\mathrm{CP}_{cl}^{\mathsf{U}} \cup \{(\mathrm{defNeg}), (\mathrm{defAnd})\}).$$

A typical derivation and a typical example in $\mathrm{CP}_{cl}^{\mathsf{U}}$:

$$\begin{aligned} \mathrm{CP}_{cl}^{\mathsf{U}} \vdash \mathsf{U} \lhd x \rhd \mathsf{U} &= (y \lhd \mathsf{U} \rhd z) \lhd x \rhd (v \lhd \mathsf{U} \rhd w) \\ &= (y \lhd x \rhd v) \lhd \mathsf{U} \rhd (z \lhd x \rhd w) \\ &= \mathsf{U}, \end{aligned}$$

$$\begin{aligned} \mathrm{CP}_{cl}^{\mathsf{U}} \vdash (\mathsf{F} \lhd a \rhd \mathsf{U}) \lhd b \rhd (\mathsf{T} \lhd a \rhd \mathsf{U}) &= (\mathsf{F} \lhd b \rhd \mathsf{T}) \lhd a \rhd (\mathsf{U} \lhd b \rhd \mathsf{U}) \\ &= (\mathsf{F} \lhd b \rhd \mathsf{T}) \lhd a \rhd \mathsf{U}. \end{aligned}$$

[6] The term "normal form" is not used in [13], but Guzmán's follow-up paper [12] states that the completeness of a Gentzen system for conditional logic "relies heavily on the axiomatization of the equational theory of C, and on the normal form in this equational theory; see [13]. The author believes that an understanding of this normal form, as presented in section 3 of [13], is essential in order to see through the formalisms in the proofs of (2.3), (2.7) and (2.10)."

This example motivates the following definitions of the extension of $\widetilde{\alpha}(P)$ to the three-valued case and of the CL-basic forms. It will be seen that $(\mathsf{F} \lhd b \rhd \mathsf{T}) \lhd a \rhd (\mathsf{U} \lhd b \rhd \mathsf{U})$ is the basic form according to the ordering $a < b$, while this is $(\mathsf{F} \lhd a \rhd \mathsf{U}) \lhd b \rhd (\mathsf{T} \lhd a \rhd \mathsf{U})$ if $b < a$. Thus, the shared alphabet of *each* of the three terms in this example must be $\{a, b\}$.

Definition 4.2 *For P a mem-basic form, its **shared alphabet** $\widetilde{\alpha}(P)$ is the following set of atoms, defined with help of the auxiliary function $h : MBF_A^{\mathsf{U}} \times 2^A \to 2^A$:*

$$\begin{aligned}
&\widetilde{\alpha}(P) = h(P, \alpha(P)),\\
&h(\mathsf{U}, x) = x,\\
&h(\mathsf{T}, x) = h(\mathsf{F}, x) = \emptyset,\\
&h(P_1 \lhd a \rhd P_2, x) = \{a\} \cup (h(P_1, x \setminus \{a\}) \cap h(P_2, x \setminus \{a\})).
\end{aligned}$$

For each $\sigma \in A^s$, we write $F_\sigma(\vec{\mathsf{U}})$ for $F_\sigma(P_0, ..., P_{2^{|\sigma|}-1})$ if for all i, $P_i = \mathsf{U}$ (where $|\sigma|$ determines the length of $\vec{\mathsf{U}}$).

Lemma 4.3 *If P is a mem-basic form with only U-leaves, then* (i) $\widetilde{\alpha}(P) = \alpha(P)$ *and* (ii) *for all $\sigma \in A^s$, $\mathrm{CP}_{cl}^{\mathsf{U}} \vdash P = F_\sigma(\vec{\mathsf{U}}) = \mathsf{U}$.*

Proof: Both (i) and (ii) follow by structural induction on P. □

Lemma 4.4 *If P is a mem-basic form and $a \in \widetilde{\alpha}(P)$, then $\mathrm{CP}_{cl}^{\mathsf{U}} \vdash P = \ell_a(P) \lhd a \rhd r_a(P)$.*

Proof: Like the proof of Lemma 3.5. □

We note that the analogue of Lemma 3.6 is not valid: if P and Q are mem-basic forms and $\mathrm{CP}_{cl}^{\mathsf{U}} \vdash P = Q$, then $\widetilde{\alpha}(P) = \widetilde{\alpha}(Q)$ does not necessarily hold, e.g $\widetilde{\alpha}(\mathsf{U}) = \emptyset$ and $\widetilde{\alpha}(\mathsf{U} \lhd a \rhd \mathsf{U}) = \{a\}$.

Definition 4.5 *A mem-basic form P is a **CLU-basic form** if for σ that represents $\widetilde{\alpha}(P)$,*

$$P = F_\sigma(P_0, ..., P_{2^{|\sigma|}-1})$$

and the following conditions are satisfied:

(i) *for some $j \in \{0, ..., 2^{|\sigma|} - 1\}$, $P_j \neq \mathsf{U}$, and*

(ii) *for all $i \in \{0, ..., 2^{|\sigma|} - 1\}$, P_i is a CLU-basic form.*

We will often write **CUbf** instead of "CLU-basic form". For example, $\mathsf{U} \lhd a \rhd \mathsf{T}$ is a CUbf. Note that the above definition implies that $F_\rho(\vec{\mathsf{U}})$ is not a CUbf for any $\rho \in A^s \setminus \{\epsilon\}$. More generally, a mem-basic form in MBF_A^{U} of depth larger than 0 with only U-leaves is not a CUbf.

Lemma 4.6 *If P is a mem-basic form and σ represents $\widetilde{\alpha}(P)$, then there exist mem-basic forms $P_0, ..., P_{2^{|\sigma|}-1}$ such that*

$$\mathrm{CP}_{cl}^{\mathsf{U}} \vdash P = F_\sigma(P_0, ..., P_{2^{|\sigma|}-1}).$$

Proof: By induction on the depth of P. The base case $d(P) = 0$ is trivial. If $d(P) > 0$ and $\sigma = a\rho$, then ρ represents the ordering of the shared alphabets of the mem-basic forms $\ell_a(P)$ and $r_a(P)$ that are both of lesser depth, hence

$$\begin{aligned} \mathrm{CP}_{cl}^{\mathsf{U}} \vdash P &= \ell_a(P) \lhd a \rhd r_a(P) && \text{(by Lemma 4.4)} \\ &= F_\rho(P_0, ..., P_{2^{|\rho|}-1}) \lhd a \rhd F_\rho(P'_0, ..., P'_{2^{|\rho|}-1}) && \text{(IH)} \\ &= F_\sigma(P_0, ..., P_{2^{|\sigma|}-1}) \end{aligned}$$

if $P_{2^{|\rho|}+i} = P'_i$ for $i = 0, ..., 2^{|\rho|} - 1$. □

Theorem 4.7 *For each mem-basic form P there is a unique CLU-basic form Q such that*

$$\mathrm{CP}_{cl} \vdash P = Q.$$

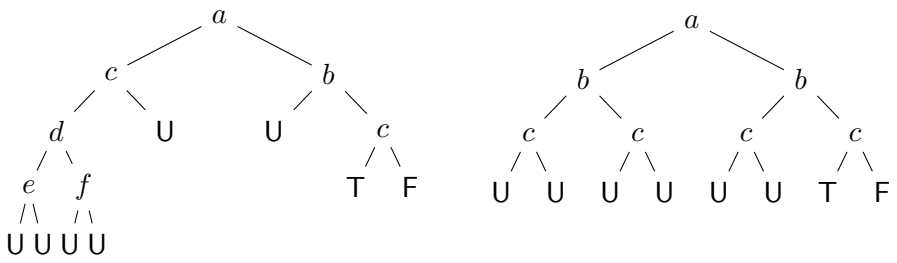
Proof: By induction on the depth of P. If $d(P) = 0$ then $P \in \{\mathsf{T}, \mathsf{F}, \mathsf{U}\}$ and we are done.

If $d(P) > 0$ then $|\widetilde{\alpha}(P)| = n$ for some $n \geq 1$. For σ that represents $\widetilde{\alpha}(P)$ it follows by Lemma 4.6 that $\mathrm{CP}_{cl}^{\mathsf{U}} \vdash P = F_\sigma(P_0, ..., P_{2^n-1})$. By induction, there are unique CUbfs Q_i such that for all i, $\mathrm{CP}_{cl}^{\mathsf{U}} \vdash P_i = Q_i$. The following two cases have to be distinguished:

1. For all i, $Q_i = \mathsf{U}$. Then $Q = \mathsf{U}$ is a CUbf, and by Lemma 4.3, it is the unique CUbf that satisfies $\mathrm{CP}_{cl}^{\mathsf{U}} \vdash P = Q$.
2. For some $j \in \{0, ..., 2^{|\sigma|} - 1\}$, $Q_j \neq \mathsf{U}$. Then $Q = F_\sigma(Q_0, ..., Q_{2^{|\sigma|}-1})$ is a CUbf and therefore it is the unique CUbf over $(\alpha(P), <)$ that satisfies $\mathrm{CP}_{cl}^{\mathsf{U}} \vdash P = Q$.

□

Example 4.8 *Consider the mem-basic form P that we display graphically below on the left, and observe that $\widetilde{\alpha}(P) = \{a, b, c\}$. If $abc \in A^s$ agrees with $(A, <)$, the CLU-basic form of P is shown on the right.*



Note that the example remains valid if in both P and its CUbf one of the leaves T or F is replaced by U. If another order of $\widetilde{\alpha}(P)$ agrees with $(A, <)$, the corresponding CUbf is easily found.

The normalisation function $mbf : \mathcal{C}_A \to \mathcal{C}_A$, defined in [4] and used in the previous section, can be easily extended to $\mathcal{C}_A^{\mathsf{U}} \to \mathcal{C}_A^{\mathsf{U}}$; we show this in Appendix A. This implies that for all $P \in \mathcal{C}_A^{\mathsf{U}}$, $\mathrm{CP}_{mem}^{\mathsf{U}} \vdash P = mbf(P)$, by which we find the following result:

Corollary 4.9 *For all $P, Q \in \mathcal{C}_A^{\mathsf{U}}$, $\mathrm{CP}_{cl}^{\mathsf{U}} \vdash P = Q$ if, and only if, $[\![P]\!]_{cl} = [\![Q]\!]_{cl}$, where $[\![P]\!]_{cl}$ is P's unique CLU-basic form.*

Corollary 4.9 and the above theorem imply that $\mathrm{CP}_{cl}^{\mathsf{U}}$ axiomatises three-valued conditional valuation congruence, that is, the congruence on $\mathcal{C}_A^{\mathsf{U}}$ defined by having equal CLU-basic forms.

It remains to be shown that CℓSCL is completely axiomatised by the equational logic CL_3.

Theorem 4.10 *For all terms s, t over $\Sigma_{\mathrm{SCL}}(A)$, $\mathrm{CL}_3 \vdash s = t \iff$ CℓSCL $\vdash s = t$.*

Proof: As the proof of Theorem 3.13, but now relying on Theorem [7, Thm.7.16]. The extensions with U in the corresponding intermediate results require $f(\mathsf{U}) = \mathsf{U}$ and $g(\mathsf{U}) = \mathsf{U}$. □

Like in the previous section, we can use CLU-basic forms to define "normal forms" in the equational theory of CL_3: for $P \in \mathcal{S}_A^{\mathsf{U}}$ we find $f(P) \in \mathcal{C}_A^{\mathsf{U}}$,

and thus, given an ordering $(A, <)$ and according to Corollary 4.9, a unique CLU-basic form $[\![f(P)]\!]_{cl}$. So $\mathrm{CP}^{\mathsf{U}}_{cl} \vdash f(P) = [\![f(P)]\!]_{cl}$ and thus

$$\mathrm{CL}_3 \vdash P = g([\![f(P)]\!]_{cl}),$$

so $g([\![f(P)]\!]_{cl})$ can be defined as the normal form of P.

5 Independent Axiomatisations of CL

Guzmán and Squier state in [13] that the seven equational axioms in Table 1, i.e.

(1.1a) $x'' = x$

(1.1b) $(x \wedge y)' = x' \vee y'$

(1.1c) $(x \wedge y) \wedge z = x \wedge (y \wedge z)$

(1.1d) $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$

(1.1e) $(x \vee y) \wedge z = (x \wedge z) \vee (x' \wedge y \wedge z)$

(1.1f) $x \vee (x \wedge y) = x$

(1.1g) $(x \wedge y) \vee (y \wedge x) = (y \wedge x) \vee (x \wedge y)$

are to the best of their knowledge independent. In this section we show that this is not the case and provide several axiomatisations that are independent. Also, we prove that $\mathrm{MSCL} \prec \mathrm{C}\ell\mathrm{SCL}_2$.

As an alternative to axiom (1.1e) that does not assume the associative law (1.1c), we consider both possible readings of (1.1e):

(Mem1) $(x \vee y) \wedge z = (x \wedge z) \vee (x' \wedge (y \wedge z))$,

(Mem2) $(x \vee y) \wedge z = (x \wedge z) \vee ((x' \wedge y) \wedge z)$.

Furthermore, we shall write CL^{U} for CL with (only) the constant U distinguished. For the axiomatisations of CL^{U}, CL_2 and CL_3 (thus, U or/and T and F are distinguished), the following axioms are provided in [13]:

(1.4a) $\mathsf{U}' = \mathsf{U}$,

(1.4b) $\mathsf{T} \wedge x = x$,

(1.4c) $\mathsf{T}' = \mathsf{F}$.

In the proofs below, we used the theorem prover *Prover9* and the tool *Mace4* for generating finite models, for both tools see [17]. Derivations from

equational axiomatisations were found with *Prover9*, and *Mace4* was used to prove all independence results. We used these tools on a Macbook Pro with a 2.4GHz dual-core Intel Core i5 processor and 4GB of RAM. We demonstrate the use of both tools in Appendix B).

Theorem 5.1 (i) *Conditional logic with no distinguished constants* (CL) *is completely axiomatised by the five axioms* (1.1a), (1.1b), (Mem1), (1.1f), (1.1g), *and this group of axioms is independent.*

(ii) *Conditional logic with no distinguished constants* (CL) *is completely axiomatised by the six axioms* (1.1a), (1.1b), (1.1d), (Mem2), (1.1f), (1.1g), *and this group of axioms is independent.*

(iii) *Conditional logic with (only)* T *and* F *distinguished* (CL_2) *is completely axiomatised by axioms* (1.4b), (1.4c), *and those in* (i) *or* (ii). *Each of these groups of axioms is independent.*

(iv) *Conditional logic with (only)* U *distinguished* (CL^{U}) *is completely axiomatised by axiom* (1.4a) *and those in* (i) *or* (ii). *Each of these groups of axioms is independent.*

(v) *Conditional logic with* T, F *and* U *distinguished* (CL_3) *is completely axiomatised by axioms* (1.4a), (1.4b), (1.4c), *and those in* (i) *or* (ii). *Each of these groups of axioms is independent.*

Proof: It suffices to show that the axioms (1.1a)–(1.1g) follow from (i) and (ii), and that both axiomatisations of CL_3 are independent.

(i) If axiom (1.1e) is replaced by (Mem1), then (1.1c) and (1.1d) are derivable from the mentioned five axioms. This follows with *Prover9*: the "Prover9 Options" `lpo` and `pass` give the best results, although the first proof is long and time-consuming and both proofs takes less time when (1.1g) is left out: first (1.1c) and with that, (1.1d): $< 56''$ and $< 6''$, respectively. This proof of (1.1c) in *Prover9* is explained in Appendix B.

(ii) If axiom (1.1e) is replaced by (Mem2) then (1.1c) is derivable from the mentioned six axioms. This follows with *Prover9* and the best result is obtained with options `lpo` and `unfold` and without (1.1g): $< 36''$.

Completeness of the axiomatisations in (i)–(v) follows from the completeness results proved in [13, Cy.2.7].

The two axiomatisations in (v) of CL_3 are independent, this quickly follows with *Mace4*. This implies that all axiomatisations in (i)–(iv) are also independent. In Appendix B we demonstrate how a simple proof of the independence of (1.1f) of the first axiomatisation in (i) can be obtained with *Mace4*, and then comment on the proof mentioned here. □

Table 4: EqCL and EqCL$^{\mathsf{U}}$, axiomatisations that are equivalent with CL_2 and CL_3, respectively

EqCL :	$\mathsf{F} = \neg\mathsf{T}$	(Neg)
	$x \mathbin{\overset{\circ}{\vee}} y = \neg(\neg x \mathbin{\overset{\circ}{\wedge}} \neg y)$	(Or)
	$\mathsf{T} \mathbin{\overset{\circ}{\wedge}} x = x$	(Tand)
	$x \mathbin{\overset{\circ}{\wedge}} (x \mathbin{\overset{\circ}{\vee}} y) = x$	(Abs)
	$(x \mathbin{\overset{\circ}{\vee}} y) \mathbin{\overset{\circ}{\wedge}} z = (\neg x \mathbin{\overset{\circ}{\wedge}} (y \mathbin{\overset{\circ}{\wedge}} z)) \mathbin{\overset{\circ}{\vee}} (x \mathbin{\overset{\circ}{\wedge}} z)$	(Mem)
	$(x \mathbin{\overset{\circ}{\wedge}} y) \mathbin{\overset{\circ}{\vee}} (y \mathbin{\overset{\circ}{\wedge}} x) = (y \mathbin{\overset{\circ}{\wedge}} x) \mathbin{\overset{\circ}{\vee}} (x \mathbin{\overset{\circ}{\wedge}} y)$	(Com)
EqCL$^{\mathsf{U}}$:	$\neg\mathsf{U} = \mathsf{U}$	(Und)

The *Prover9* results mentioned above can be obtained slightly faster if (1.1f) is replaced by its dual $x \wedge (x \vee y) = x$ and/or if (Mem1) is replaced by its symmetric counterpart $(x \vee y) \wedge z = (x' \wedge (y \wedge z)) \vee (x \wedge z)$. We note that each of these replacements preserves independence. Below, we discuss an alternative axiomatisation based on these observations.

We return to our preferred notation $\overset{\circ}{\wedge}, \overset{\circ}{\vee}$ and $\neg$. In Table 4, we extend EqMSCL to EqCL, and EqMSCL$^{\mathsf{U}}$ to EqCL$^{\mathsf{U}}$ with the axiom (Com). Notice the differences between axiom (Or) and $\neg(x \mathbin{\overset{\circ}{\wedge}} y) = \neg x \mathbin{\overset{\circ}{\vee}} \neg y$ (i.e. axiom (1.1b) in preferred notation), and between axiom (Mem) and the axioms (Mem1) and (Mem2) (the two possible readings of (1.1e) in preferred notation).

Theorem 5.2 (i) *Conditional logic with no constants distinguished* (CL) *is completely axiomatised by the five axioms* $\neg\neg x = x$ *and* (Or), (Abs), (Mem), *and* (Com) *from Table 4. Moreover, these axioms are independent.*

(ii) *Conditional logic with* T *and* F *distinguished* (CL_2) *is completely axiomatised by the six axioms of* EqCL *(see Table 4). Moreover, these axioms are independent.*

(iii) *Conditional logic with only* U *distinguished* (CL$^{\mathsf{U}}$) *is completely axiomatised by the six axioms* $\neg\neg x = x$, $\neg\mathsf{U} = \mathsf{U}$, *and* (Or), (Abs), (Mem) *and* (Com) *in Table 4. Moreover, these axioms are independent.*

(iv) *Conditional logic with all constants distinguished* (CL_3) *is completely axiomatised by the seven axioms of* EqCL$^{\mathsf{U}}$ *(see Table 4). Moreover, these axioms are independent.*

Proof: It follows easily (either by hand or with help of *Prover9*) that the axioms in each of (i)–(iv) both imply those of the associated axiomatisation in Theorem 5.1 that uses the axioms in Thm.5.1(i), *and* are their consequences. This proves the mentioned completeness results.

The independence of the axiomatisations (iii) and (iv) follows quickly with *Mace4*, and implies the independence of (i) and (ii), respectively. □

Corollary 5.3 $\mathrm{MSCL} \prec \mathrm{C\ell SCL}_2$ *and* $\mathrm{MSCL}^{\mathsf{U}} \prec \mathrm{C\ell SCL}$*, i.e., for all terms* s,t *over* $\Sigma_{\mathrm{SCL}}(A)$*,* $\mathrm{MSCL} \vdash s = t \implies \mathrm{C\ell SCL}_2 \vdash s = t$*, and for all terms* s,t *over* $\Sigma^{\mathsf{U}}_{\mathrm{SCL}}(A)$*,* $\mathrm{MSCL}^{\mathsf{U}} \vdash s = t \implies \mathrm{C\ell SCL} \vdash s = t$*.*

Proof: EqMSCL axiomatises MSCL and by Theorem 5.2(ii), it is a proper subset of the independent axiomatisation EqCL of $\mathrm{C\ell SCL}_2$. Hence, $\mathrm{MSCL} \prec \mathrm{C\ell SCL}_2$.

Similarly, $\mathrm{EqMSCL}^{\mathsf{U}}$ axiomatises $\mathrm{MSCL}^{\mathsf{U}}$ and is by Theorem 5.2(iv) a proper subset of the independent axiomatisation $\mathrm{EqCL}^{\mathsf{U}}$ of $\mathrm{C\ell SCL}$. Hence, $\mathrm{MSCL}^{\mathsf{U}} \prec \mathrm{C\ell SCL}$. □

6 Discussion and Conclusions

We studied conditional logic and provided for its variants with distinguished constants T and F, and with or without U, a detailed explanation of its 'basic forms', and included these variants in our framework of short-circuit logics. In this section we address two questions: which logic is defined by conditional logic when restricting to the full left-sequential connectives and negation, and what does it mean to define SCLs without the constants T and F. We end with some conclusions and remarks on related work.

Full left-sequential evaluation is commutative in CL, consequences. Let $\mathrm{CL}_3(\mathbin{\overset{\bullet}{\wedge}}, \mathbin{\overset{\bullet}{\vee}})$ denote the extension of CL_3 with the connectives $\mathbin{\overset{\bullet}{\wedge}}$ and $\mathbin{\overset{\bullet}{\vee}}$ and their defining axioms

$$x \mathbin{\overset{\bullet}{\wedge}} y = (x \mathbin{\overset{\circ}{\wedge}} y) \mathbin{\overset{\circ}{\vee}} (y \mathbin{\overset{\circ}{\wedge}} x) \quad \text{and} \quad x \mathbin{\overset{\bullet}{\vee}} y = \neg(\neg x \mathbin{\overset{\bullet}{\wedge}} \neg y).$$

Then $\mathrm{CL}_3(\mathbin{\overset{\bullet}{\wedge}}, \mathbin{\overset{\bullet}{\vee}})$ incorporates Bochvar's *three-valued logic* [8]. In [1], this logic is denoted as $\mathbb{S}_3$ and it is proved that $\mathbb{S}_3$ is completely axiomatised by the axioms in Table 5.[7]

[7] The missing axiom (S5) reads $x \mathbin{\bullet\!\!\rightarrow} y = \neg x \mathbin{\overset{\bullet}{\vee}} y$, but the connective $\bullet\!\!\rightarrow$ (full left-sequential implication) is irrelevant here. We note that the axioms (S1)–(S4),(S6)–(S9),(S11) are independent (according to *Mace4*) and that (S10) is a consequence of (S3),(S4),(S6)–(S9) (according to Prover9).

Table 5: Axioms of Bochvar's logic, taken from [1]

(S1)	$\neg\mathsf{T} = \mathsf{F}$
(S2)	$\neg\mathsf{U} = \mathsf{U}$
(S3)	$\neg\neg x = x$
(S4)	$\neg(x \mathbin{\overset{\bullet}{\wedge}} y) = \neg x \mathbin{\overset{\bullet}{\vee}} \neg y$
(S6)	$x \mathbin{\overset{\bullet}{\wedge}} (y \mathbin{\overset{\bullet}{\wedge}} z) = (x \mathbin{\overset{\bullet}{\wedge}} y) \mathbin{\overset{\bullet}{\wedge}} z$
(S7)	$\mathsf{T} \mathbin{\overset{\bullet}{\wedge}} x = x$
(S8)	$x \mathbin{\overset{\bullet}{\vee}} (\neg x \mathbin{\overset{\bullet}{\wedge}} y) = x \mathbin{\overset{\bullet}{\vee}} y$
(S9)	$x \mathbin{\overset{\bullet}{\wedge}} y = y \mathbin{\overset{\bullet}{\wedge}} x$
(S10)	$x \mathbin{\overset{\bullet}{\wedge}} (y \mathbin{\overset{\bullet}{\vee}} z) = (x \mathbin{\overset{\bullet}{\wedge}} y) \mathbin{\overset{\bullet}{\vee}} (x \mathbin{\overset{\bullet}{\wedge}} z)$
(S11)	$\mathsf{U} \mathbin{\overset{\bullet}{\wedge}} x = \mathsf{U}$

With *Prover9* it quickly follows that these axioms are consequences of $\mathrm{CL}_3(\overset{\bullet}{\wedge}, \overset{\bullet}{\vee})$. Moreover, in $\mathrm{CL}_2(\overset{\bullet}{\wedge}, \overset{\bullet}{\vee})$ and $\mathrm{CL}(\overset{\bullet}{\wedge}, \overset{\bullet}{\vee})$, the corresponding version of the axiomatisation of $\mathbb{S}_3$, i.e.

(S1), (S3), (S4), (S6)–(S9) and (S3), (S4), (S8)–(S10), respectively,

is also derivable (the latter set implies (S6)). The two-valued variant of $\mathbb{S}_3$, say $\mathbb{S}_2$, satisfies

$$\mathbb{S}_2 \prec \text{ “propositional logic with } \neg, \overset{\bullet}{\wedge} \text{ and } \overset{\bullet}{\vee}\text{”}$$

because the absorption law $x \mathbin{\overset{\bullet}{\wedge}} (x \mathbin{\overset{\bullet}{\vee}} y) = x$ and its commutative variants (and duals) do not hold in $\mathbb{S}_2$ (a counter-model is quickly found by *Mace4*).

Short-circuit logics without constants. Obviously, all two-valued SCLs can be defined without the constants T and F. In the case of FSCL, this significantly reduces its axiomatisation, a complete set of axioms is $\neg\neg x = x$, $x \mathbin{\overset{\circ}{\vee}} y = \neg(\neg x \mathbin{\overset{\circ}{\wedge}} \neg y)$, and $(x \mathbin{\overset{\circ}{\wedge}} y) \mathbin{\overset{\circ}{\wedge}} z = x \mathbin{\overset{\circ}{\wedge}} (y \mathbin{\overset{\circ}{\wedge}} z)$ (as noted in [18]). All other axioms of FSCL involve (sub)terms with only F-leaves (or with only T-leaves), such as $x \mathbin{\overset{\circ}{\wedge}} \mathsf{F} = \neg x \mathbin{\overset{\circ}{\wedge}} \mathsf{F}$, and these cannot be expressed in FSCL without T and F. Hence, the full left-sequential connectives are not definable in FSCL without T and F.

Let MSCL_0 be defined as MSCL, but without the constants T and F (which are not definable). This is similar to the way CL is set up, leaving open the choice to adjoin T and F, and U. From the axiomatisation of CL_2 in Table 4 (Theorem 5.2(ii)), a complete axiomatisation of MSCL_0, say EqMSCL_0, is obtained by omitting the axiom (Com) from those listed in Theorem 5.2(i).[8] Obviously, EqMSCL_0 is independent because the original axiomatisation is. Moreover, full left-sequential conjunction is definable in MSCL_0 by

$$x \mathbin{\underset{\bullet}{\wedge}} y = (x \mathbin{\overset{\circ}{\vee}} (y \mathbin{\underset{\circ}{\wedge}} \neg y)) \mathbin{\underset{\circ}{\wedge}} y \quad \text{or} \quad x \mathbin{\underset{\bullet}{\wedge}} y = (x \mathbin{\underset{\circ}{\wedge}} y) \mathbin{\overset{\circ}{\vee}} (y \mathbin{\underset{\circ}{\wedge}} x).$$

We conclude that MSCL_0 compared to MSCL imposes no constraints other than the undefinability of T and F.

Finally, we note that $\mathrm{MSCL}_0 \vdash (x \mathbin{\overset{\circ}{\vee}} \neg x) \mathbin{\underset{\circ}{\wedge}} x = x$, $(\neg x \mathbin{\overset{\circ}{\vee}} x) \mathbin{\underset{\circ}{\wedge}} x = x$, the equations that are the counterparts of the axiom $\mathsf{T} \mathbin{\underset{\circ}{\wedge}} x = x$ (Tand), so the question is indeed what would be missed when using MSCL_0 "in practice". We also note that omitting (1.1g) from each of the two axiomatisations of CL_2 in Theorem 5.1 does not yield a complete axiomatisation of MSCL_0: *Mace4* finds a counter-model for $(\neg x \mathbin{\overset{\circ}{\vee}} x) \mathbin{\underset{\circ}{\wedge}} x = x$, and for $x \mathbin{\overset{\circ}{\vee}} \neg x = \neg x \mathbin{\overset{\circ}{\vee}} x$.

For the case of $\mathrm{MSCL}_0^{\mathsf{U}}$ (with U and the axiom $\neg\mathsf{U} = \mathsf{U}$), similar observations apply.

Conclusions. Starting from Guzmán and Squier's conditional logic, we introduced the two-valued short-circuit logic $\mathrm{C}\ell\mathrm{SCL}_2$, axiomatised by CL_2, and three-valued $\mathrm{C}\ell\mathrm{SCL}$, axiomatised by CL_3. Then we analysed the commutativity of the full left-sequential connectives in CL_3 and found that these and negation define Bochvar's three-valued logic. Finally, we investigated the independence of the equational axiomatisations of CL, and paid attention to short-circuit logic without constants. We conclude with two remarks.

(1) Three equations that hold in MSCL_0 and express properties related to falsehood are

$$\begin{aligned} &x \mathbin{\underset{\circ}{\wedge}} \neg x = \neg x \mathbin{\underset{\circ}{\wedge}} x, \\ &(x \mathbin{\underset{\circ}{\wedge}} \neg x) \mathbin{\overset{\circ}{\vee}} x = x, \text{ and} \\ &(x \mathbin{\underset{\circ}{\wedge}} \neg x) \mathbin{\underset{\circ}{\wedge}} y = x \mathbin{\underset{\circ}{\wedge}} \neg x. \end{aligned}$$

Another equation that relates to falsehood and does not hold in MSCL, but does hold in CL, is

$$(x \mathbin{\underset{\circ}{\wedge}} \neg x) \mathbin{\overset{\circ}{\vee}} (y \mathbin{\underset{\circ}{\wedge}} \neg y) = (y \mathbin{\underset{\circ}{\wedge}} \neg y) \mathbin{\overset{\circ}{\vee}} (x \mathbin{\underset{\circ}{\wedge}} \neg x).$$

[8] $\mathrm{EqMSCL} \vdash s = t \Rightarrow \mathrm{EqMSCL}_0 \vdash s = t$ follows by induction on the length of derivations.

Furthermore, $x \mathbin{\overset{\circ}{\wedge}} \neg x = y \mathbin{\overset{\circ}{\wedge}} \neg y$ is *not* a consequence of CL_2 (by *Mace4*). These facts support our conjecture that there exists no short-circuit logic without additional constants in between CL_2 and SSCL (i.e., the SCL that represents propositional logic in the signature $\{\overset{\circ}{\wedge}, \overset{\circ}{\vee}, \neg, \mathsf{T}, \mathsf{F}\}$). We note that adding the axiom $x \mathbin{\overset{\circ}{\wedge}} \neg x = y \mathbin{\overset{\circ}{\wedge}} \neg y$ to CL_2 (or $(x \mathbin{\overset{\circ}{\wedge}} \neg x) \mathbin{\overset{\circ}{\vee}} y = y$, or $x \mathbin{\overset{\circ}{\wedge}} \mathsf{F} = \mathsf{F}$) yields SSCL (by *Prover9*). Of course, with $a \in A$ and $|A| > 1$, the ad-hoc SCL defined by $\mathrm{CL}_2 \cup \{(a \mathbin{\overset{\circ}{\wedge}} \neg a) \mathbin{\overset{\circ}{\vee}} x = x\}$ is in between CL_2 and SSCL.

(2) The equation $(x \mathbin{\overset{\circ}{\wedge}} \mathsf{U}) \mathbin{\overset{\circ}{\vee}} \mathsf{U} = \mathsf{U}$, which follows immediately from CL^{U} and CL_3, can be seen as a concise characterisation of the difference between CL^{U} and $\mathrm{MSCL}_0^{\mathsf{U}}$, and of the difference between CL_3 and $\mathrm{MSCL}^{\mathsf{U}}$.

Related work and application perspective. In [20], Staudt introduced evaluation trees and thereby axiomatised *Free Fully Evaluated Logic* (FFEL), a sublogic of FSCL that has (only) full left-sequential connectives (and, like FSCL, is immune to side effects):

$$\begin{aligned}
&\text{(FEL1)} && \mathsf{F} = \neg\mathsf{T}\\
&\text{(FEL2)} && x \mathbin{\overset{\bullet}{\vee}} y = \neg(\neg x \mathbin{\overset{\bullet}{\wedge}} \neg y)\\
&\text{(FEL3)} && \neg\neg x = x\\
&\text{(FEL4)} && (x \mathbin{\overset{\bullet}{\wedge}} y) \mathbin{\overset{\bullet}{\wedge}} z = x \mathbin{\overset{\bullet}{\wedge}} (y \mathbin{\overset{\bullet}{\wedge}} z)\\
&\text{(FEL5)} && \mathsf{T} \mathbin{\overset{\bullet}{\wedge}} x = x\\
&\text{(FEL6)} && x \mathbin{\overset{\bullet}{\wedge}} \mathsf{T} = x\\
&\text{(FEL7)} && x \mathbin{\overset{\bullet}{\wedge}} \mathsf{F} = \mathsf{F} \mathbin{\overset{\bullet}{\wedge}} x\\
&\text{(FEL8)} && x \mathbin{\overset{\bullet}{\wedge}} \mathsf{F} = \neg x \mathbin{\overset{\bullet}{\wedge}} \mathsf{F}\\
&\text{(FEL9)} && (x \mathbin{\overset{\bullet}{\wedge}} \mathsf{F}) \mathbin{\overset{\bullet}{\vee}} y = (x \mathbin{\overset{\bullet}{\vee}} \mathsf{T}) \mathbin{\overset{\bullet}{\wedge}} y\\
&\text{(FEL10)} && x \mathbin{\overset{\bullet}{\vee}} (y \mathbin{\overset{\bullet}{\wedge}} \mathsf{F}) = x \mathbin{\overset{\bullet}{\wedge}} (y \mathbin{\overset{\bullet}{\vee}} \mathsf{T})
\end{aligned}$$

These axioms are consequences of $\mathbb{S}_2$ (according to Prover9), but FFEL is obviously weaker: $x \mathbin{\overset{\bullet}{\wedge}} y = y \mathbin{\overset{\bullet}{\wedge}} x$ does not hold. We note that FSCL and FFEL were axiomatised for the first time in [20], and that FFEL with undefinedness ($\mathrm{FFEL}^{\mathsf{U}}$) is axiomatised in [19], while an equational axiomatisation of FSCL with undefinedness ($\mathrm{FSCL}^{\mathsf{U}}$) has only been conjectured [7, Conject.8.1].

Also in [7], we mentioned the following attractive challenge: *Find a convincing example that distinguishes* $\mathrm{MSCL}^{\mathsf{U}}$ *from Conditional Logic as defined in Guzmán and Squier (1990).*

Our current research on "fracterm calculus for partial meadows" [5] proves to be such a convincing example. A partial meadow is a field with

a partial division function for which division by 0 is undefined. Fracterm calculus for partial meadows aims to formalise elementary arithmetic including division by defining a three-valued sequential first-order logic, on top of a three-valued short-circuit logic with sequential, noncommutative connectives, where a *partial equality* relation $=_{\mathsf{p}}$ is used to construct atoms, e.g. $\mathsf{x} =_{\mathsf{p}} 0$. In the Tarski semantics for this first-order logic, CℓSCL is sound and thus the right short-circuit logic to use. An equation that holds in CℓSCL and not in MSCL$^{\mathsf{U}}$ was mentioned above in Conclusions, remark (2):

$$(a \mathbin{\diagup\!\!\!\wedge} \mathsf{U}) \mathbin{\diagup\!\!\!\vee} \mathsf{U} = \mathsf{U}.$$

Now the requested example is found if we replace a with an atom that always evaluates to either *true* or *false*, such as $\mathsf{x} =_{\mathsf{p}} 0$, and U with a partial equality that always evaluates to *undefined*, such as $\frac{1}{0} =_{\mathsf{p}} 0$. In the context of calculi for partial data types, we plan to study similar applications of CℓSCL.

Acknowledgements. We are grateful to the reviewers for their comments, corrections and questions, which were very helpful and have led to several improvements.

References

[1] Jan A. Bergstra, Inge Bethke, and Piet Rodenburg. A propositional logic with 4 values: true, false, divergent and meaningless. *Journal of Applied Non-Classical Logics*, 5(2):199–217, 1995. doi:10.1080/11663081.1995.10510855.

[2] Jan A Bergstra, Jan Heering, and Paul Klint. Module algebra. *Journal of the ACM*, 37(2):335–372, 1990. doi:10.1145/77600.77621.

[3] Jan A. Bergstra and Alban Ponse. Proposition algebra. *ACM Transactions on Computational Logic*, 12(3), 2011. doi:10.1145/1929954.1929958.

[4] Jan A. Bergstra and Alban Ponse. Evaluation trees for proposition algebra, 2017. arXiv:1504.08321.

[5] Jan A. Bergstra and Alban Ponse. Fracterm calculus for partial meadows, 2025. arXiv:2502.13812.

[6] Jan A. Bergstra, Alban Ponse, and Daan J. C. Staudt. Short-circuit logic, 2013. (v4, first version v1 appeared in 2010). arXiv:1010.3674.

[7] Jan A. Bergstra, Alban Ponse, and Daan J. C. Staudt. Non-commutative propositional logic with short-circuit evaluation. *Journal of Applied Non-Classical Logics*, 31(3-4):234–278, 2021. doi:10.1080/11663081.2021.2010954.

[8] Dmitry Anatolévich Bochvar. Ob odnom tréhznacnom isčislenii i égo priménénii k analiza paradoksov klassičékogo râssirénnogo funkcional'nogo isčislénid. *Matématičeskij Sbornik*, 4:287–308, 1939. *Translation*: Dmitry Anatolévich Bochvar and Merrie Bergmann (1981). On a three-valued logical calculus and its application to the analysis of the paradoxes of the classical extended functional calculus. *History and Philosophy of Logic*, 2(1-2), 87-112. doi:10.1080/01445348108837023.

[9] Alonzo Church. Conditioned disjunction as a primitive connective for the propositional calculus. *Portugaliae Mathematica*, 7:87–90, 1948. URL: https://purl.pt/2171/1/j-5293-b-vol7-fasc2-art3_PDF/j-5293-b-vol7-fasc2-art3_PDF_01-B-R0300/j-5293-b-vol7-fasc2-art3_0000_capa1-90_t01-B-R0300.pdf.

[10] Alonzo Church. *Introduction to Mathematical Logic*. Princeton University Press, 1956. Also published as a paperback edition. De Gruyter Brill, 1991. doi:10.1515/9781400881451.

[11] David Gries. *The Science of Programming*. Texts and Monographs in Computer Science. Springer, 1981. doi:10.1007/978-1-4612-5983-1.

[12] Fernando Guzmán. A Gentzen system for conditional logic. *Studia Logic*, 53:243–257, 1994. doi:10.1007/BF01054712.

[13] Fernando Guzmán and Craig C. Squier. The algebra of conditional logic. *Algebra Universalis*, 27:88–110, 1990. doi:10.1007/BF01190256.

[14] Charles Antony Richard Hoare. A couple of novelties in the propositional calculus. *Mathematical Logic Quarterly*, 31(9-12):173–178, 1985. doi:10.1002/malq.19850310905.

[15] John McCarthy. A basis for a mathematical theory of computation. In Paul Braffort and David Hirschberg, editors, *Computer Programming and Formal Systems*, volume 35 of *Studies in Logic and the Foundations of Mathematics*, pages 33–70. Elsevier, 1963. doi:10.1016/S0049-237X(08)72018-4.

[16] John McCarthy. Predicate calculus with "undefined" as a truth-value, 1963. Stanford Artificial Intelligence Project Memo I (5 pages). URL:

https://stacks.stanford.edu/file/druid:zc129sh9123/SC1041_SAIL_AIM_001.pdf.

[17] William McCune. The GUI: Prover9 and Mace4 with a graphical user interface. Prover9-Mace4 Version 0.5B, 2008. Prover9-Mace4-v05B.zip, March 14, 2008. URL: https://www.cs.unm.edu/~mccune/prover9/gui/v05.html.

[18] Alban Ponse and Daan J. C. Staudt. An independent axiomatisation for free short-circuit logic. *Journal of Applied Non-Classical Logics*, 28(1):35–71, 2018. doi:10.1080/11663081.2018.1448637.

[19] Alban Ponse and Daan J. C. Staudt. Fully evaluated left-sequential logics, 2024. arXiv:2403.14576.

[20] Daan J. C. Staudt. Completeness for two left-sequential logics, 2012. arXiv:1206.1936.

A A Normalisation Function for mem-basic Forms

The two normalisation functions $bf()$ for two-valued basic forms and $mbf()$ for mem-basic forms, both defined in [4], can be easily lifted to the three-valued setting.

Define the basic form function $bf : \mathcal{C}_A^{\mathsf{U}} \to BF_A^{\mathsf{U}}$ and, given $Q, R \in BF_A^{\mathsf{U}}$, the auxiliary function $[\mathsf{T} \mapsto Q, \mathsf{F} \mapsto R] : BF_A^{\mathsf{U}} \to BF_A^{\mathsf{U}}$ for which postfix notation $P[\mathsf{T} \mapsto Q, \mathsf{F} \mapsto R]$ is used, as follows:

$$\begin{aligned} &bf(B) = B \ \text{ for } B \in \{\mathsf{T}, \mathsf{F}, \mathsf{U}\}, && \mathsf{T}[\mathsf{T} \mapsto Q, \mathsf{F} \mapsto R] = Q, \\ &bf(a) = \mathsf{T} \lhd a \rhd \mathsf{F} \ \text{ for } a \in A, && \mathsf{F}[\mathsf{T} \mapsto Q, \mathsf{F} \mapsto R] = R, \\ &bf(P \lhd Q \rhd R) = bf(Q)[\mathsf{T} \mapsto bf(P), \mathsf{F} \mapsto bf(R)], && \mathsf{U}[\mathsf{T} \mapsto Q, \mathsf{F} \mapsto R] = \mathsf{U}, \end{aligned}$$

and

$$(P_1 \lhd a \rhd P_2)[\mathsf{T} \mapsto Q, \mathsf{F} \mapsto R] = P_1[\mathsf{T} \mapsto Q, \mathsf{F} \mapsto R] \lhd a \rhd P_2[\mathsf{T} \mapsto Q, \mathsf{F} \mapsto R].$$

From [7, App.A8], it follows that for all $P \in \mathcal{C}_A^{\mathsf{U}}$, $\mathrm{CP}^{\mathsf{U}} \vdash P = bf(P)$, and for all $P, Q \in \mathcal{C}_A^{\mathsf{U}}$, $\mathrm{CP}^{\mathsf{U}} \vdash P = Q$ if, and only if, $bf(P) = bf(Q)$.

Next, define the mem-basic form function $mbf : \mathcal{C}_A^{\mathsf{U}} \to \mathcal{C}_A^{\mathsf{U}}$ by $mbf(P) = mf(bf(P))$, and the auxiliary function $mf : BF_A^{\mathsf{U}} \to BF_A^{\mathsf{U}}$ by $mf(B) = B$ for $B \in \{\mathsf{T}, \mathsf{F}, \mathsf{U}\}$ and $mf(P \lhd a \rhd Q) = mf(\ell_a(P)) \lhd a \rhd mf(r_a(Q))$, with $\ell_a()$ and $r_a()$ as in Definition 2.4.

It follows easily that $mbf()$ is a normalisation function yielding membasic forms: the extension to $\mathcal{C}_A^{\mathsf{U}}$ of Theorem 5.9 in [4], i.e.

$$\text{For all } P \in \mathcal{C}_A^{\mathsf{U}}, \mathrm{CP}_{mem} \vdash P = mbf(P),$$

and of the supporting lemmas is trivial: all inductive proofs in [4] require one additional, trivial base case (for U).

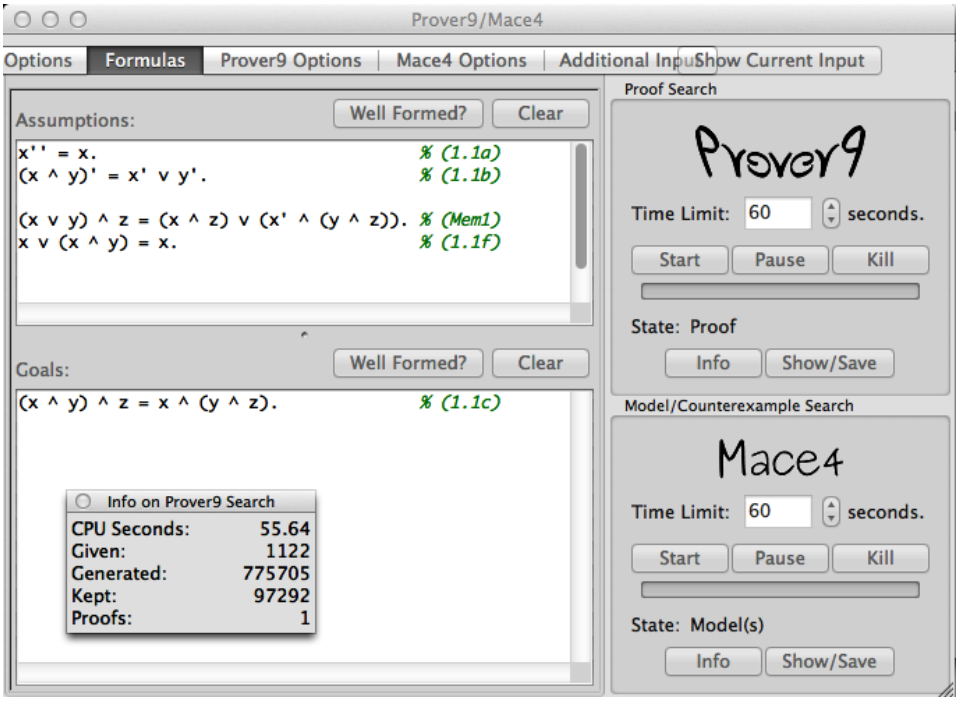
B Prover9 and Mace4 Results

We explain how we used *Prover9* and *Mace4* [17] to prove Theorem 5.1(i).

First, we show how axiom (1.1c) follows in *Prover9* from the four axioms

$$(1.1a),\ (1.1b),\ (\text{Mem1}),\ (1.1f),$$

which are the "Assumptions" in the screen "Prover9/Mace4", the graphical user interface of [17], and (1.1c) is the only goal listed under "Goals". The "Info on Prover9 Search" screen was activated while a proof was being searched for and was slid over the interface window after termination (a proof was found):



Below we show an edited version of the output as displayed in the screen "Prover9 Proof". To obtain a concise representation, we used the screen tool "Reformat ..." with the options `parents_only` and `Renumber Proof`, where the first option only provides source references and omits further explanations. The output was then saved and shortened by reducing the line length by 2 characters, omitting the lines `[8 ... 1020]`, and adding a line break and spaces in the section "`...= prooftrans =...`" :

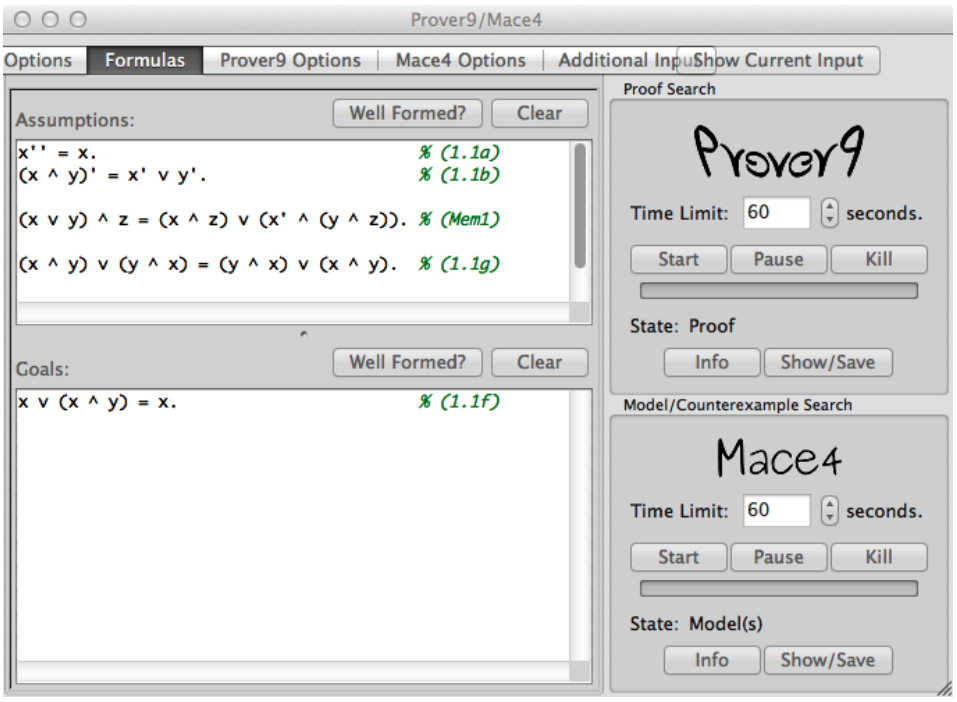
```
============================== prooftrans ============================
Prover9 (32) version Dec-2007, Dec 2007.
Process 1215 was started by alban on Albans-MacBook-Pro.local,
Fri Sep 19 14:17:42 2025
The command was "/Applications/Prover9-Mace4-v05B.app/Contents/
                                    Resources/bin-mac-intel/prover9".
============================== end of head ===========================

============================== end of input ==========================

============================== PROOF =================================

% -------- Comments from original proof --------
% Proof 1 at 55.62 (+ 0.66) seconds.
% Length of proof is 1023.
% Level of proof is 120.
% Maximum clause weight is 56.
% Given clauses 1122.

1 (x ^ y) ^ z = x ^ (y ^ z) # label(non_clause) # label(goal).  [].
2 x'' = x.  [].
3 (x ^ y)' = x' v y'.  [].
4 (x v y) ^ z = (x ^ z) v (x' ^ (y ^ z)).  [].
5 (x ^ y) v (x' ^ (z ^ y)) = (x v z) ^ y.  [4].
6 x v (x ^ y) = x.  [].
7 (c1 ^ c2) ^ c3 != c1 ^ (c2 ^ c3).  [1].

[8 ... 1020]

1021 x ^ ((x' v y) ^ z) = (x ^ y) ^ z.  [1019,1020,999].
1022 (x ^ y) ^ z = x ^ (y ^ z).  [1014,1021,1013].
1023 $F.  [1022,7].

============================== end of proof ==========================
```

Secondly, we show how one can use *Mace4* to demonstrate that ax-

iom (1.1f) is independent of the four axioms

$$\text{(1.1a)},\ \text{(1.1b)},\ \text{(Mem1)},\ \text{(1.1g)}.$$

In the graphical user interface, these four axioms are taken as assumptions and (1.1f) as the goal. Then running *Mace4* with the default options generates a finite model that satisfies these assumptions and provides a counterexample for this goal. A screenshot of the graphical user interface for this case:



The complete file saved from the output window "Mace4 Model", which provides a model that refutes axiom (1.1f):

```
interpretation( 2, [number = 1,seconds = 0], [
    function(^(_,_), [
        1,1,
        1,1]),
    function(v(_,_), [
        1,1,
        1,1]),
    function('(_), [0,1]),
    function(c1, [0]),
    function(c2, [0])]).
```

This generated model has domain $\{0,1\}$ (indicated by the `2` in the first line) and defines each application of either `^` or `v` to return `1`, and negation `'` as the identity. The given counterexample is obtained by instantiating (1.1f) with `x = y = 0` (`c1` and `c2`). We note that it is easy to see that axioms (1.1a), (1.1b), (Mem1) and (1.1g) are satisfied in this model, as is the congruence rule.

By repeating this approach for each of the other axioms, one can prove that (1.1a), (1.1b), (Mem1), (1.1f) and (1.1g) form an independent axiomatisation, as required.

Finally, we note that in the proof of Theorem 5.1(i), this independence result is a consequence of a more general result, namely the independence of the first axiomatisation of CL_3 given in Theorem 5.1(v): the eight axioms

(1.1a), (1.1b), (Mem1), (1.1f), (1.1g) and (1.4a), (1.4b), (1.4c)

are independent. This follows from eight slightly more complex models and counterexamples, all of which were quickly generated by *Mace4*. Consequently, the first five axioms are independent.

© Scientific Annals of Computer Science 2025